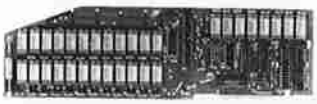
Nr. 6/85 Juni
DM 6,50, sfr 6,50, öS 50, Lit 5900, hfl 7,50

PEEKER

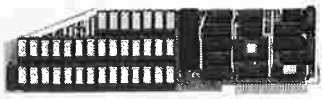
MAGAZIN FÜR APPLE-COMPUTER

Hires-Farbbit
Superdump
Fourier-Analyse
Ampersand-Utilities
Pascal-RAM-Disk
Solitaire
Diversi-DOS
Lazar

Hüthig
PUBLIKATION

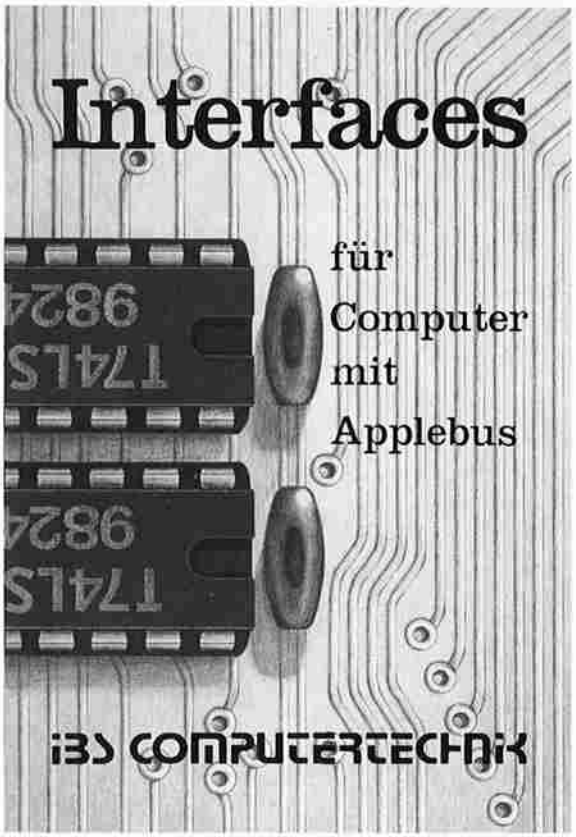


AP 13 und AP 17
RAM-Karten zum Einsatz als Pseudodisk unter CP/M, USCD und APPLE-DOS. Speichergröße von 64 kByte bis 256 kByte.
Bestell-Nr.: A 1013 a-b
A 1017 a-d



AP 33
RAMDISK der neuen Generation. Für besonders speicherintensive Arbeiten ist der Ausbau in Stufen von 64 kByte bis 1MByte möglich.
Bestell-Nr. A 1033

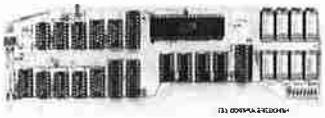
AP 14
Floppy-Controller für alle Anwendungsfälle. 10 Laufwerke können gleichzeitig angeschlossen werden. 4 × 8" DSDD, 4 × 5¼" DSDD und zwei Apple-Standardlaufwerke. Maximal ca. 10MByte im Direktzugriff.
Bestell-Nr.: A 1014



AP 19
12-Kanal AD-DA-Wandler mit 12 bit Auflösung und 25 μ sec Wandlungszeit. Eingangsspannung ±10 V. Ein schneller Wandler für extrem schnelle Anwendungen.
Bestell-Nr.: A 1019

NEU! jetzt 512 k-RAM

AP 20
INTEMEX mit 68 000 CPU und 128 k-RAM. Diese Karte macht aus Ihrem Rechner mit „Applebus" einen echten 16 bit-Rechner. Eine Zusatzkarte (AP 26) ermöglicht einen Arbeitsspeicher bis zu einem MByte und an Software gibt es einiges. Z.B. stehen drei Betriebssysteme und die wichtigsten Hochsprachen zur Verfügung.
Bestell-Nr. A 1020



NEU! 8 MHz Takt

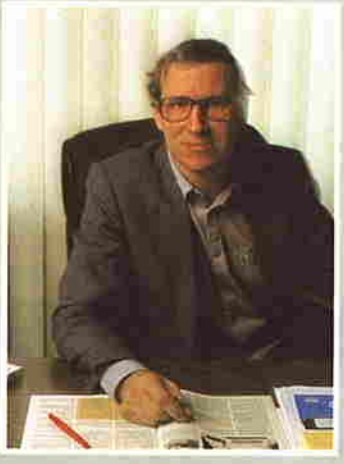
AP 22
INTEMEX mit Z 80 B-CPU und 64 k-RAM. Wenn Sie einmal diese Karte in Aktion gesehen haben, werden Sie auch feststellen: „Geschwindigkeit ist keine Hexerei, man braucht nur die AP 22". Mit dieser Karte wird Ihr APPLE II zum z.Z. schnellsten CP/M-Computer, und in Verbindung mit dem SPACE 84 erhalten Sie Computerleistung, die wirklich einmalig ist. Wir vermitteln gerne eine Vorführung.
Bestell-Nr. A 1022

NEU!!!

Das Interface-Buch von IBS, ein Buch für Alle, die Ihren APPLE II oder Kompatiblen optimal nutzen wollen. Detaillierte Schaltpläne, Bauteilelisten und Benutzungshinweise zu allen IBS-Interfaces finden Sie jetzt in einem Buch vereint. Ausführliche Abhandlungen über Spezialschaltungen, über Anwendungsmöglichkeiten, über neue Softwarewelten aber auch über die Grenzen des APPLE II-Systems bestimmen den Wert dieses Buches.
Für nur DM 8,00 erhalten Sie dieses Buch ab sofort bei Ihrem Computerfachhändler oder für DM 8,00 + DM 2,00 Versandkosten bei IBS COMPUTERVERTRIEB.

IBS COMPUTERTECHNIK

Olper Straße 10 · 4800 Bielefeld 14 · Tel.: 0521/444032 · W. Germany
1011 Rose Marie Lane 16 · Stockton CA 95207 · Tel. 209/473–7473 USA



EDITORIAL

Jeder Apple-Besitzer, gleich ob er kleine oder große Programme schreibt, kommt zwangsläufig im Laufe der Zeit zu der schmerzlichen Erkenntnis, daß es fehlerlose Programme nicht gibt. There is always one more bug! Folglich erfordert jedes kommerziell vertriebene Programm einen gewissen „Support" zur Ausmerzung von Fehlern. Zwei Beispiele:

Bug 1: In meinem Programm „RAMDISKLC" aus Peeker, Heft 1/85, S. 8, ist ein sehr unerquicklicher, aber leider nicht sofort evidenter Bug, der in exakt folgendem Fall auftritt:

```
10 PRINT CHR$ (4) "OPEN ZAHLEN-
   TEXTFILE"
20 PRINT CHR$ (4) "WRITE ZAHLEN-
   TEXTFILE"
30 PRINT Z1: PRINT Z2...: PRINT ZN:
   REM usw.
```

Es muß also ein Textfile (kein Binärfile usw.) eröffnet werden; ferner muß ein Schreibvorgang stattfinden (WRITE, nicht READ); und schließlich müssen Zahlen (nicht Strings) geschrieben werden. Wenn nunmehr eine Zahl zwei Textfile-Sektoren überlappt, enthält genau diese Nahtstelle zwischen vorangehendem und folgendem Sektor „Zahlenschrott".

Bug 2: Die ProDOS-Version 1.0.1 (und 1.0.2) in Verbindung mit der BASIC.SYSTEM-Version 1.0 hat einen ähnlichen Bug (s. S. 52 in diesem Heft):

```
10 PRINT CHR$ (4) „OPEN TEXTFILE"
20 PRINT CHR$ (4) „READ TEXTFILE,F1"
30 INPUT ZAHL: REM oder INPUT STRING
```

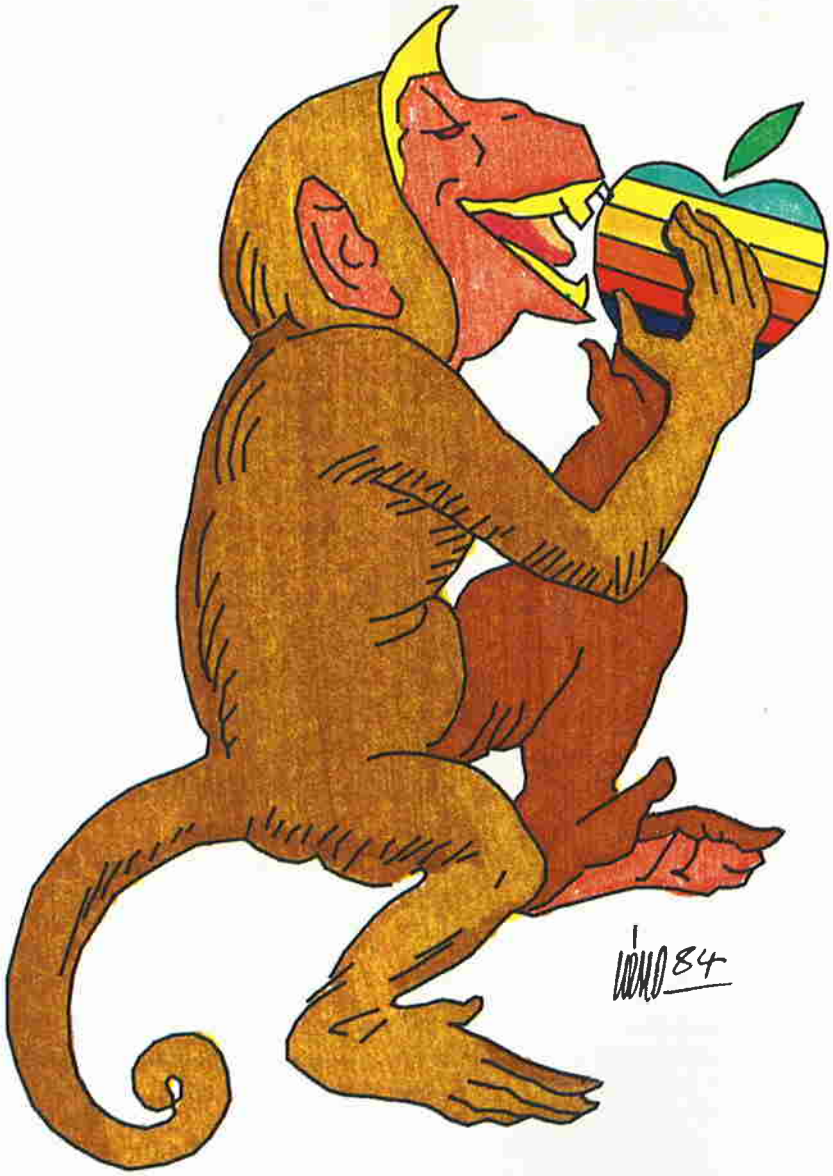
Es muß also ein Textfile eröffnet werden (kein Binärfile usw.); ferner muß von dem Textfile gelesen werden (READ, nicht WRITE); und schließlich muß eine Feldnummer F > = 1 spezifiziert werden. Dagegen ist es gleichgültig, ob eine Zahl oder ein String eingelesen wird. Hierzu kommt es jedoch gar nicht mehr, denn in Zeile 30 weist das BASIC.SYSTEM den Variablenwert nicht der INPUT-Variablen zu, sondern schreibt ihn erbarmungslos in den Speicher des Applesoft-Programms (ab $0801), das somit zerstört wird.

Zur Würdigung des „Supports" muß man den Preis der Software mit berücksichtigen. Nehmen wir einmal der Einfachheit halber an, daß ein Peeker-Heft im Durchschnitt 10 Progamme enthält. Da ein einzelnes Heft DM 6,50 kostet, hat jedes einzelne Programm einen Durchschnittspreis von 6,5 : 10 = 0,65 DM. Unser „Support" für das 65-Pfennig-Programm beschränkt sich daher auf die Veröffentlichung eines Patches, der den RAMDISKLC-Bug beseitigt.

ProDOS wird anstelle von DOS 3.3 heute in Verbindung mit der Hardware geliefert. Ein halbwegs komplettes Hardware-Paket mit Laufwerk(en) usw. kostet bei einem Vertragshändler heute immer noch gut DM 4000,–. Dafür könnte man über 50 (in Worten fünfzig!) Jahre lang den Peeker abonnieren. Welchen „Support" erhält man jedoch in bezug auf den ProDOS-Bug? In der Regel gar keinen, weil man aus naheliegenden Gründen kaum einen Vertragshändler findet, der so tief in dieses Betriebssystem eingedrungen ist, daß er eine Patentlösung („Patch") anbieten kann. Denn schließlich ist ein Händler ein Händler und kein Entwickler.

Ulrich Stiehl

Hüthig
PUBLIKATION



So entstand der APPLE-Biß!
(Frei nach Wilhelm Busch) Computergrafik von N. G. Barbieri

INHALT

6/85

Impressum
Peeker
Magazin für Apple-Computer
2. Jahrgang 1985
ISSN 0176–9200
© für den gesamten Inhalt einschließlich der Programme Dr. Alfred Hüthig Verlag, Heidelberg 1985

Verleger und Herausgeber:
Dipl.-Kfm. Holger Hüthig
Geschäftsführung Zeitschriften:
Heinz Melcher
Chefredakteur:
Ulrich Stiehl (us) Tel. (0 62 21) 48 93 52
(Bitte nur in redaktionellen Angelegenheiten anrufen)

Anzeigenleitung:
Jürgen Maurer, Tel. (0 62 21) 48 92 18
z. Zt. gilt Anzeigenpreisliste Nr. 3
Vertriebsleitung:
Ruth Biller, Tel. (0 62 21) 48 92 80
Produktionsleitung: Gunter Sokollek
Gestaltung: Rainer Schmitt
Titelbild: Creative Computer Service, Mannheim

peeker

MAGAZIN FÜR APPLE-COMPUTER

Impressum 5
Inserentenverzeichnis 51
Leserbriefe 69
Wie gibt man Maschinenprogramme ein? 70
Extended Graphic System 71
Terminal-Programm für die Super-Serial-Card 72

Testberichte
Diversi-DOS 4-C 72
Lasar IIze 75
Typenradschreibmaschine Brother CE 50 76

Editierhilfen für den Apple GPLE und DOUBLE TAKE 77
Lernprogramme von INTUS 78

6 **Graf-quattro**
Teil 2: Schneller als der Schall
von Nino G. Barbieri

16 **Das Farbbit**
Pseudo-Double-Hires auf dem Apple II Plus
von Hans-Detlef Siewert

22 **Superdump**
Das universelle Hires-Grafik-Dump-Programm
von Jürgen Geiß

34 **Hardcopy auf Olympia ESW 100 RO**
Ein Typenraddrucker wird grafikfähig
von Ralf Menssen

38 **Fourier-Analyse**
von Peter Walber

43 **Applesoft-Erweiterungen**
Eine Sammlung von Ampersand-Routinen
von Markus Enzinger

48 **RAM-Disk-Driver für Pascal 1.1**
von Michael Schröter

52 **ProDOS**
Theorie und Praxis
von Ulrich Stiehl

55 **Apple-CP/M**
Mit einem RAM-Disk-Driver für die 64K-Karte
von Karl-Walter Bott

64 **Solitaire**
und wie man es löst
von Stefan Maas

Verlag:
Dr. Alfred Hüthig Verlag GmbH
Im Weiher 10, Postfach 10 28 69
6900 Heidelberg
Telefon (0 62 21) 4 89-0
Telex 4-6 17 27 hued d.

Erscheinungsweise: 12 Hefte jährlich,
Erscheinungstag jeweils 1 Woche vor Monatsbeginn.
Jahresabonnement DM 72,–, einschließlich MwSt,
im Inland portofrei. Einzelheft DM 6,50
Vertrieb Handel:
MZV – Moderner Zeitschriften Vertrieb GmbH
Breslauer Str. 5, Postfach 1123,
8057 Eching b. München,
Tel. 089/319 10 67, Telex 0 522 656

Zahlungen: an den Dr. Alfred Hüthig Verlag GmbH, D-6900 Heidelberg 1: Postscheckkonten: BRD: Karlsruhe 485 45-753; Österreich: Wien 755588 88; Schweiz: Basel 40-24417; Niederlande: Den Haag 1 457 28; Italien: Mailand 47718; Belgien: Brüssel 723026; Dänemark: Kopenhagen 349 69; Norwegen: Oslo 994 24; Schweden: Stockholm 5477 76-5

Bankkonten: Landeszentralbank Heidelberg 67 207 341; BLZ 672 000 00; Deutsche Bank Heidelberg 0 2165 041; BLZ 672 700 03; Bezirkssparkasse Heidelberg 204 51, BLZ 672 500 20.

Herstellung: Heidelberger Verlagsanstalt
Printed in Germany

Graf-quattro

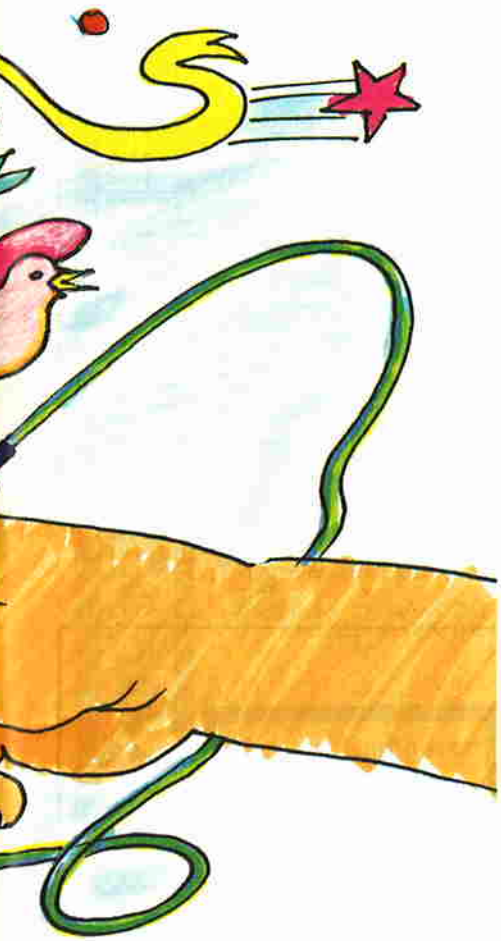
von N.G. Barbieri

Teil 2: Schneller als der Schall

Jedes Hires-Grafikprogramm hat selbstverständlich einen Hires-Cursor. Ich halte nicht viel von *einem* Cursor! Für mich sind zwei Cursoren viel besser.
Warum zwei Cursoren?
– Weil der Mensch zwei Hände hat,
– eine Linie zwei Enden,
– ein Zirkel zwei Spitzen,
– et cetera.

1. CURSOR1 und CURSOR2

Die ersten zwei Utilities dieses Beitrags bilden die zwei Cursoren auf der Hires-Seite und sind – wie die Programme aus dem ersten Teil dieser Serie (Peeker, 4/85) – selbständige Assemblerprogramme, die direkt von Applesoft oder anderen Maschinenprogrammen aufgerufen werden können. Im Grunde genommen sind die zwei Routinen identisch bis auf die Form der Shapes: „+" für CURSOR1 und „x" für CURSOR2. Bevor ich auf die Funktion eingehe, einige Worte über die Applesoft-Routine HFNS ($F6B9). Diese untersucht die aktuelle Programmzeile, holt sich einen ersten Wert, der nicht kleiner als 0 und nicht größer als 279 sein darf, prüft, ob ein Komma vorhanden ist, und holt sich dann einen zweiten Wert, der nicht kleiner als 0 und nicht größer als 191 sein darf. Am Ende sind die Werte, die wir X und Y nennen wollen, jeweils im X-Register (XLOW), im Y-Register (XHIGH) und im A-Register (Y-Wert). Damit ist eine bequeme Art und Weise gegeben, von Applesoft aus einen direkten Anschluß an Assemblerroutinen für die Hires-Grafik mit Übergabe von Werten, Variablen oder Formeln für die X-Y-Koordinaten zu finden.



Die einzelne Cursor-Routine ist so geschrieben, daß beim ersten CALL ein Shape mit XDRAW an den gegebenen Koordinaten gebildet wird. Bei weiteren CALLs wird immer zuerst das alte Shape gelöscht und dann ein neues gebildet, gegebenenfalls auf einer neuen X-Y-Position. Damit wird die spätere Arbeit in Applesoft wesentlich erleichtert. Für die Cursoren werden die normalen Applesoft-Shapes verwendet, aber direkt ohne den Umweg über das offset-indizierte Verfahren. Das geschieht, indem zuerst im X- und Y-Register jeweils das Low- und High-Byte der Speicherstelle, wo die Shape-Vektoren anfangen, und im A-Register die Rotation (in unserem Fall immer 0) geladen werden und dann ein JSR auf XDRAW ($F65D) erfolgt. Ist ein DRAW gewünscht, gelten dieselben Regeln, nur sollte dann der JSR auf $F601 geschehen. Möchte man diese Grundroutine auch für andere Zwecke benutzen (vielleicht für ein Flugzeug in einem Computerspiel), so braucht man nur andere Shape-Vektoren an der gegebenen Stelle einzufügen.

Um die Cursoren zu steuern, hier die entsprechenden CALLs:
CURSOR1: Das erste Mal CALL 24769, danach CALL 24757.
CURSOR2: Das erste Mal CALL 24820, danach CALL 24808.
Die entsprechenden BASIC-Befehle lauten:
CALL 24757: X, Y (später)
CALL 24769: X, Y (anfangs)
für CURSOR1 und
CALL 24808: X, Y (später)
CALL 24820: X, Y (anfangs)
für CURSOR2.
Ähnlich wie bei XPLOT (s. Teil 1 dieser Serie) darf man nicht den syntaktischen „:" zwischen CALL und Koordinatenwerten bzw. Variablen vergessen. Die Koordinaten werden danach automatisch gespeichert, und zwar jeweils in der Reihenfolge XLOW, XHIGH und Y für CURSOR1 in $03CA, $03CB, $03CC und für CURSOR2 in $03CD, $03CE und $03CF. Damit ist eine einfache und sichere Übergabe der Koordinaten für die anderen Utilities gewährleistet.

2. LINIE

Die Tatsache, daß die Cursor-Koordinaten immer verfügbar sind, erlaubt es, mit einer sehr einfachen Routine eine Linie zwischen den zwei Punkten zu ziehen. Dadurch, daß wir jetzt auch über einen XPLOT-Modus verfügen (s. Teil 1), ist es möglich, die Linie auch blinken zu lassen und wie ein Gummiband zu behandeln. Ein Flag bei $0300 (768) regelt dies. Ist die Speicherstelle auf 0, so wird die Linie in XPLOT gebildet, sonst erfolgt der normale HPLOT. Die Assemblerroutine erklärt sich selbst. Von Applesoft aus reicht einfach ein
CALL 24861.
Die Unterroutine CURSORE sorgt dafür, daß die Cursoren veschwinden oder neu gebildet werden, je nach Bedarf. Damit wird verhindert, daß Linien oder andere Gebilde und Cursoren sich gegenseitig XORen.

3. VIERECK

Benutzt man die zwei Cursoren als diagonal entgegengesetzte Ecken, ist es sehr einfach, ein Viereck zu bilden. Die Technik ist ähnlich der Routine LINIE, hat aber zusätzlich eine Unterroutine DOLINE, die auch für die folgenden Programme nützlich sein wird. Von Applesoft aus wird VIERECK mit
CALL 24927
aufgerufen.

4. BOX

Die BOX-Routine erlaubt es, ein Viereck zu füllen. Um die Sache interessanter zu machen, wird hier alternativ in vertikaler Richtung mit zwei Farben gefüllt, wobei auch eine Farbe 8 berücksichtigt wird, d.h. XPLOT. Dafür verwenden wir zwei Pointer, und zwar $0302 (770) und $0303 (771), wo jeweils die gewünschten Farben gepokt werden müssen. Möchte man nur mit einer Farbe füllen, poke man auf beiden Stellen denselben Wert und rufe danach BOX mit
CALL 25007
auf. Will man zum Beispiel den Inhalt des Vierecks negativ darstellen, dann muß man zweimal die 8 poken. Die Unterroutine SETCOL sorgt dafür, daß entweder der entsprechende COLORMASK-Wert zum

APPLE – DISKETTEN LAUFWERKE – APPLE

Original Disk II m. Controller + DOS 3.3 + Hdbuch	DM	995,–
Original Disk II (2. Laufwerk)	DM	765,–
Duo-Disk Station Slimline, 2 x 143KB Chinon	DM	995,–
Siemens Disk-Laufw., 143KB, m. Kabel + Gehäuse	DM	525,–
Abor-Disk-Laufw., Slimline, 143KB, m. Kab. + Geh.	DM	435,–
Chinon Slimline, 143KB, Superleise, m. Kab. + Geh.	DM	495,–
Teac 55F, 1 MB unf. Kapazität, Shugartbus	DM	625,–
Teac 55F, kpl. in Gehäuse, 40/80 Track, 1MByte anschlußfertig, mit Umschaltung 40/80 + Kabel	DM	698,–
Zweitlaufwerk für Apple II C, 143 KB, mit Kabel	DM	498,–

APPLE – INTERFACES + MAINBOARDS – APPLE

Disk Controller f. 2 Original o. kompat. Drives	*DM	89,–
Super-Controller f. 2x Teac 55F, mit Software	**DM	186,–
80 Zeich-Karte mit 64K RAM + Softsw. f. IIe	DM	395,–
80 Zeich-Karte II, m. Softswitch, 2 Zeichensätze	*DM	159,–
16K RAM Erweiterung für II und kompatible	*DM	98,–
Z 80A Interface Karte für CPM 2.2	*DM	88,–
Z 80B Interface Karte für CPM 3.0, m. 64K RAM	DM	595,–
Printer-Grafik Interface, Epson kompatible	*DM	98,–
Centronic-Parallell Text-Interface Karte	*DM	98,–
Printer-Grafik Interface, NEC/ITOH kompatibel	*DM	169,–
Anschluß Kabel f. Paralell + Grafik Interface	DM	34,–
Buffer Grafik-Interface, benutzbar für Drucker Epson/Nec/Itoh/Okidata u. andere m. 32K Buffer	DM	348,–
Buffer Interface wie vor aber mit 64K Buffer	**DM	595,–
Anschluß Kabel f. Buffer Interface Karte	DM	39,–
232C Serielle Interface Karte	*DM	109,–
Super Serielle Interface Karte, Fullduplex	**DM	288,–
Sprach (Speech) Karte f. Sprachwiedergabe	*DM	78,–
6522 Paralell Interface Karte	*DM	149,–
Clock Karte (Datum/Uhrzeit) Ein/Ausgabe	*DM	129,–
Eprom Writer Karte (2716 - 2764)	*DM	129,–
IEEE-488 Interface Karte	*DM	398,–
Logo-Karte mit Diskette und Handbuch	DM	498,–
Musik Karte m. Diskette und Handbuch	*DM	119,–
PAL Color Interface Karte (UHF + Video)	*DM	129,–
RGB Interface Karte (f. Apple II + II+)	*DM	169,–
Wild Karte (kopiert über RAM-Bereich)	*DM	119,–
128K RAM Erweiterungs Karte m. Patchsoftware	**DM	448,–
256K RAM Erweiterungs Karte m. Patchsoftware	**DM	698,–
6809 Prozessor Exell-9 Interfacekarte	+ DM	398,–
IC-Tester Interface Karte (RAMS/TTL 54/74)	DM	448,–
Hauptplatine 48K, o. Firmware Eproms, 8 Slots	DM	548,–
Hauptplatine 64K, wie vor	DM	648,–

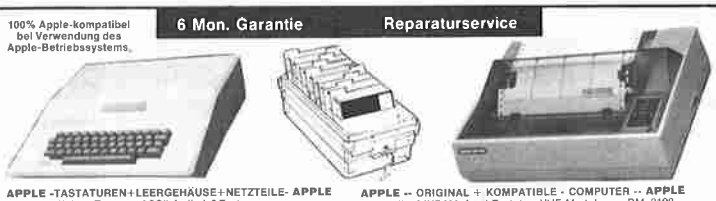
APPLE – LEERPLATINEN – APPLE

Leerplatinen der obigen Interface Karten sind alle in vergoldet, m. Bestück-Aufdruck + Best.-Plan lieferbar.

Leerplatinen mit der Kennzeichnung	**DM	39,–
Leerplatinen mit der Kennzeichnung	*DM	24,50
Leerplatinen mit der Kennzeichnung	+ DM	33,–
Experimentier Platine, für Apple Slot EX300	DM	29,90
Leerplatine Motherboard, 48K, mit Best.-Aufdruck	DM	66,–
Leerplatine Motherboard, 64K, mit 6502 + Z80	DM	99,–

100% Apple-kompatibel bei Verwendung des Apple-Betriebssystems.

6 Mon. Garantie **Reparaturservice**



APPLE -TASTATUREN+LEERGEHÄUSE+NETZTEILE- APPLE

Standard Einbau-Tastatur, ASCII, freibel. 9 Tasten, Doppelbelg. aller Tasten, Groß+Kleinschr. Nr. N26	DM	139,–
Tastatur wie vor, jedoch mit 15er Block Nr. N67	DM	178,–
Tastatur Nr. N67, mit sep. Gehäuse, kpl. m. Kabel	DM	249,–
Seperate Tastatur IBM-Look, anschlußfertig mit Kabel, Dopp. belegt. Tasten, frei prog. Funkt. Tasten	DM	298,–
Seperate Tastatur, Eprom progr.-bar, Cursorblock Tastenfeld mit 24 Funktionen, Deutsch o. ASCII	DM	398,–
Leergehäuse Standard, passend f. Tastatur Nr. N26	DM	98,–
dto. wie vor, jedoch passend für 15er Tast. N67	DM	119,–
Leergehäuse wie Mewa 9000, für Einbau von zwei Slimline-Laufwerken + Tastaturanschluß, Plastic	DM	159,–
Leergehäuse wie vor, jedoch in Metall-Ausführung	DM	198,–
Leergehäuse IBM-Look, 1, 2 Laufw. Standard 5"	DM	198,–
Leergehäuse für 1 Slimline Laufwerk, Metall	DM	19,90
Leergehäuse f. Duo-Disk = 2x Slimline-Laufwerke	DM	98,–
Leergehäuse f. Duo-Disk = 2x Standard-Laufw.	DM	109,–
Leergehäuse f. Duo-Disk = 2x Slimline + Netzteil	DM	179,–
Schaltnetzteile für APPLE u. kompatible Rechner		
+5V/3,5A -5V/0,5A +12V/2A -12V/0,5A N73	DM	99,80
+5V/5 A -5V/0,5A +12V/2,5A -12V/0,5A N74	DM	119,80
+5V/7,5A -5V/0,5A +12V/2A -12V/0,5A N75	DM	149,50

********* – DISKETTEN + DISKETTEN-BOXEN – *********

Profi Disk Box, m. Schloß, Klars.-Deckel, 100 Disks	DM	49,–
Disk Box mit Klars.-Deckel, ca. 70 Disketten 5"	DM	26,–
Hartbox für 10 Disketten, mit Klappdeckel	DM	6,90

Disketten, mit Verstarkungsring, 3 Jahre Garantie

ABX 100 = SS/SD 10 Stck. a. 3,98 50 Stck. a. 3,88 500 Stck. a. 3,59		
ABX 200 = SS/DS 10 Stck. a. 4,08 50 Stck. a. 3,98 500 Stck. a. 3,69		
ABX 400 = DS/DD 10 Stck. a. 4,98 50 Stck. a. 4,88 500 Stck. a. 4,59		

APPLE – ORIGINAL + KOMPATIBLE - COMPUTER – APPLE

APPLE IIe, 64KRAM, Ascii-Tastat. + UHF-Modul.	DM	2198,–
APPLE IIe, Einstiegspaket = Computer 64KRAM, + Philips-Monitor 18 Mhz grün + Siemens F122 - Laufwerk 143K m. Controller + Lerndiskette	DM	3098,–
APPLE II C, 128 KRAM, ASCII Tastatur	DM	2795,–
APPLE II C, wie vor, jedoch deutsche Tastatur	DM	2998,–
MACINTOSH System komplett m. Mouse + Tastatur + Macwrite/Macpaint, Systemguide-Diskette	DM	5995,–
MEWA-48K, Apple kompat., Netzt. 5Amp, Tastatur mit Dopp-Belegung + 10 frei Prog. Tasten, mit UHF-Mod., ohne Firmware Eproms.	DM	888,–
MEWA-64K wie vor jedoch mit 15er Block	DM	948,–
MEWA 9000er Serie, mit seper. Tastat. DIN o. ASCII m. dopp. belegt. Funkt-Tasten, Cursorbl. + 15er Block Netzt. 5 Amp., Gehäuse für 2x Slimline-Laufwerke UHF-Modulator, o. Firmw.-Proms (12KROM), 48K RAM	DM	1098,–
MEWA 9000-64-C, wie vor jedoch 64K RAM	DM	1198,–
MEWA 9000-64-C Einstiegspaket, wie vor, jedoch mit 1 Disk Drive eingeb. + Disk-Contr. + Monitor, grün	DM	2098,–
IBM-Look Gehause für MEWA 9000 Serie Aufpreis	DM	66,–

********* – MATRIX- UND TYPENRAD - DRUCKER – *********

CP-80 Matrix Drucker, Epson kompat., vollgrafikf.	DM	739,–
CP-80 X mit eingeb. Interface für VC 64 I, Sim. B.	DM	899,–
SPEEDY 100-80, Epson kompat., 100 Z/s, v.-grafikf.	DM	759,–
GEMINI 10X, Neu mit Textsp., 100 Z/s, 9x9 Mat.	DM	998,–
GEMINI 15X, wie 10X, jedoch bis 375 mm Papierbr.	DM	1198,–
ITOH 8510 B, Die Zuverlässigkeit in Person	DM	1475,–
JUKI 6100, Typenraddr., 22 Z/s, TA-Typenrader	DM	1495,–

IBM * KOMPATIBEL ***** IBM *** KOMPATIBEL ***

Alle Teile unterliegen einer sorgfältigen Endkontrolle und wir übernehmen daher volle Garantie für die Funktionsfähigkeit, sowohl für die gelieferten Leerplatinen, als auch für die fertig bestückten Boards und Interface Karten, die fast ohne Ausnahme mit gesockelten IC's geliefert werden.

IBM – Kompatible Interface Karten und Mainboards – IBM

Disk Controller Karte für 2 Disk-Drives	*KL-2020	DM	298,50
Color Video Board, Farb- und Video Ausg.	*KL-2050	DM	598,–
Monochrome Video Boards	*KL-2060	DM	398,–
RS 232 Drucker Interface Karte	KL-2074	DM	199,–
Centronics Paralell Interface Karte	KL-2072	DM	198,–
Multifunktion Karte, RAM-Bereich, RS232, Paralell, Clock, mit O K bestückt	*KL-2040	DM	499,–
dto. wie vor, jedoch mit 128K bestückt	*KL-2041	DM	699,–
dto. wie vor, jedoch mit 256K bestückt	*KL-2042	DM	899,–
512K RAM Karte, mit O K bestückt	KL-2095	DM	348,–
dto. wie vor, mit 128K bestückt	KL-2096	DM	548,–
Multi I/O Interface Board, RS232, Disk-Controller, Centronics, Clock, Joystick und Lightpen-Port	*KL-2071	DM	699,50
Eprom Writer Karte (bis 128K benutzbar)			a. A.
Hardisk/Winchester Host-Adapter Karte	*KL-2022	DM	1188,–
Mainboard XT Version, 8 Slots 64K bestückt mit 1 Boot-Eprom, 6 Eprom-Plätze frei	*KL-1004	DM	996,–
dto. wie vor, jedoch mit 128K bestückt	*KL-1005	DM	1096,–
dto. wie vor, jedoch mit 256K bestückt	*KL-1006	DM	1296,–
Mainboard PC Version, 8 Slots, 64K bestückt mit 1 Boot-Eprom, 6 Eprom-Plätze frei	*KL-1010	DM	896,–
dto. wie vor, jedoch mit 128K bestückt	*KL-1011	DM	996,–
dto. wie vor, jedoch mit 256K bestückt	*KL-1012	DM	1296,–

Bei den mit * gekennzeichneten Artikeln gehört zum Lieferumfang ein Manual, und z. T. ein Schaltbild.

IBM – Kompatible Leerplatinen + Boards – IBM

Disk Controller f. 2 Drives	DM	58,–
Color Grafik Video Board	DM	78,–
RS232 Interface Karte	DM	68,–
Centronics Paratell Interface Karte	DM	68,–
Multifunktion Interface Board	DM	99,–
Eprom Writer Interface Karte	DM	78,–
Multi I/O Board	DM	78,–
Mainboard XT Version, 8 Slots	DM	98,–
Mainboard PC Version, 8 Slots	DM	96,–
RAM Card für Speicher-Erweiterung	DM	68,–

Alle Leerplatinen werden mit Bestückungsplan geliefert.
Schaltbilder, soweit verfügbar, Lieferung gegen Aufpreis.

IBM – Gehäuse/Netzteile/Tastaturen/Diskdrives – IBM

Rechner Leergehäuse f. Einbau v. 2 Drives	DM	236,–
Profi-Tastatur, m. Kabel im Gehäuse, ASCII	DM	395,–
dto. wie vor, jedoch deutsche DIN Belegung	DM	489,–
Standard Tastatur, IBM-Look, ASCII	DM	298,–
Netzteil, mit Ventilator, 100 Watt	DM	298,–
dto. wie vor jedoch 135 Watt	DM	338,–
Disketten Laufwerk, 2x40 Track DS/360 KB	DM	498,–
Harddisk 10MByte, kpl. m. Xebec Controller	DM	2990,–

Sonderliste gegen Rückporto, Katalog gegen 3 DM in Briefmarken

Computer-Artikel Nachnahmeversand unfrei, Zwischenverkauf vorbehalten.
Angebote freibleibend unter Anerkennung unserer Lieferbedingungen. Technische Änderungen vorbehalten. *Apple ist eingetragenes Warenzeichen der Fa. Apple-Computer Inc., Kalifornien. Ware mit Rückgaberecht, besonders gekennzeichnet, muß frei zurückgeschickt werden. *IBM* ist eingetragenes Warenzeichen der Firma IBM GmbH Ffm.

COMPUTER CENTER CONEX
5650 SOLINGEN 11 - Postfach 11 02 06 -11 G
Telefon (0 21 22) 7 54 49

+

ERICH-WILLI MEYER
6343 FROHNHAUSEN - Postfach 70 11
Telefon (0 27 71) 3 50 71

Für IIc und IIe mit 64K-Karte

SUPERPLOT

Double-Hires-Utility

von Karl-Walter Bott, 1984, Programmdiskette und Manual, DM 48,–

SUPERPLOT ist eine neue, ungewöhnlich kompakte und schnelle Ampersand-Utility für Double Hires, die einschließlich eines vollständigen ASCII-Shape-Zeichensatzes wahlweise in Bank 1 oder Bank 2 der Language Card liegt und damit sowohl unter ProDOS als auch unter DOS 3.3, falls letzteres in die LC-Bank geschoben wurde, benutzt und in eigene Applesoftprogramme integriert werden kann. SUPERPLOT unterstützt die üblichen HGR-Befehle, denen lediglich ein & vorangestellt werden muß, also z. B. & HPLOT 500, 100 TO 500, 150 usw. SUPERPLOT ist speziell für das Plotten von beschrifteten wissenschaftlichen Funktionskurven mit hoher Auflösung gedacht und weniger für HGR-Spiele.

Hüthig Software Service · Postfach 10 28 69 · 6900 Heidelberg 1

Ablage Bearbeiten Größe Verlag Zeichensatz Stil Hilfsmittel

pandabooks

Bismarckstr. 67, D-1000 Berlin 12 (030) 342 88 00

Gratis!
AppleQuick
Das Apple II-Brevier für häufig gebrauchte Adressen und Befehle! 64 Seiten über Applesoft, DOS, Pascal, CP/M. Gleich anfordern!

Pandabooks!

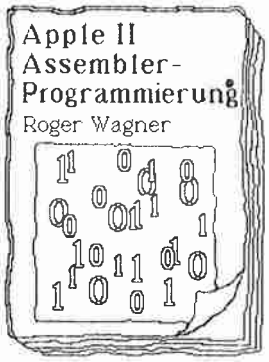
Die Qualität kommerzieller Arcadespiele läßt sich mit APPLESOFT BASIC alleine nicht erreichen. Jeffrey Stanton führt in die Eigenarten der hochauflösenden Apple-Grafik ein und präsentiert schließlich eine Reihe extrem schneller Assembler-Routinen, mit denen Sie viele Effekte bekannter Spiele selbst programmieren können. Gute BASIC-Kenntnisse werden vorausgesetzt, eine Einführung in Assembler-Programmierung wird gegeben.

3-89058-006-8 299 Seiten DM 49,--
komplett mit Disk DM 89,--

Don Worth und Peter Lechner sind die Autoren von "Beneath Apple DOS", der Bibel aller DOS 3.3-Benutzer. In ihrem neuen Buch haben sie sich ProDOS vorgenommen und mit der ihnen eigenen Gründlichkeit zerlegt. Ausführliche Beschreibung der Arbeitsweise, der Dateitypen, des "MLI" und des "BASIC SYSTEM". Das Standardwerk für jeden ProDOS-Anwender.

3-89058-036-X ca. 250 S. DM 46,--

Das Assembler-Lehrbuch für BASIC-Kenner.
Roger Wagner, der Autor vieler bekannter Software-Pakete, schrieb eine monatliche Kolumne über Assembler-Programmierung in der Apple-Zeitschrift SOFTALK.
Der vorliegende Band faßt diese Reihe, korrigiert und erweitert, zusammen: Eine stufenweise Einführung in die Befehle und Strukturen der 6502-Assemblersprache, mit vielen Beispielen von der einfachen Tongenerierung bis zum Diskettenzugriff.
3-89058-003-3 277 Seiten DM 48,--
komplett mit Disk DM 89,--

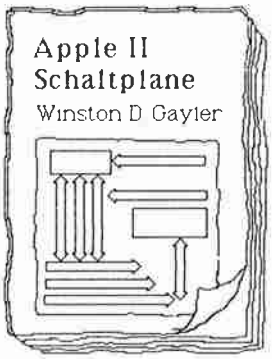


Ein Simulationsprogramm, das Sie in das Innere des 6502 Mikroprozessors führt. Sie sehen auf dem Bildschirm, wie die einzelnen Instruktionen in Zeitlupe ausgeführt werden, wie sich die Register und die Flags verändern. Ein unverzichtbares Hilfsmittel beim Erlernen der Assembler-Programmierung, danach ein wertvolles Werkzeug beim Testen Ihrer eigenen Programme. Komplett mit einem 6502 Editor/-Assembler, deutschem Handbuch (150 Seiten) und über 30 Beispielprogrammen.

3-89058-019-X DM 129,--

Die beiden Autoren haben zusammen mehr als 50 Jahre Amateurfunk-Erfahrung und präsentieren hier mehr als 20 BASIC-Programme, die jeden Funkamateur interessieren: QTH-KENNER, DÄMMERUNGSLINIE, DX-RECHNER, DOPPEL-FAHNDER, NACHWEISSCHREIBER, ALLZWECK-CONTEST-LOGGER, FIELD-DAY-LOGGER, WETTBEWERB-LOGGER u.v.a.m. Das Buch enthält die kompletten Listings für Apple II und C64.

3-89058-027-0 ca. 256 S. DM 48,--
komplett mit Disk DM 89,--



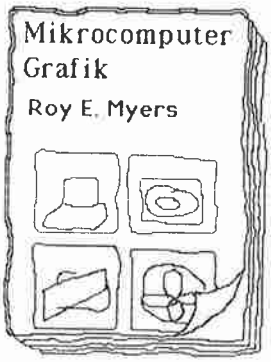
Eine detaillierte Beschreibung der Apple II und Apple IIplus Schaltungen. Wenn Sie Ihren Apple selbst reparieren, Interface-Karten oder Schaltungserweiterungen entwerfen oder einfach nur besser über das Innenleben Ihres Apple Bescheid wissen wollen - dieses Buch bietet Ihnen eine Fülle an Informationen, Schaltpläne und Zeitdiagramme, Theorie und praktische Tips.

3-89058-012-2 215 S. A4 DM64,--

Endlich anspruchsvolle Apple-Grafik für BASIC-Programmierer. Mikrocomputer Grafik

- enthält fast 80 lauffertige BASIC-Programme, die die beschriebenen Konzepte illustrieren.
- beschreibt "Hidden Line"- und "Hidden Surface"-Techniken, Skalierung, Rotation und Translation von Grafiken
- bietet eine Einführung in die Animationstechnik

3-89058-000-9 292 Seiten DM 49,--
komplett mit Disk DM 89,--



22 Pascal-Programme, mit denen Sie die Grafik-Möglichkeiten Ihres Apple voll ausschöpfen:
"Designer" ermöglicht es Ihnen, eigene Zeichensätze zu entwerfen.
"Gredit" unterstützt Sie beim Entwerfen komplexer Bildschirm-Grafiken
"Printfoto" bringt Ihre Entwürfe aufs Papier.
Darüberhinaus bietet das Buch eine Fülle fertiger Prozeduren, die Sie zeitsparend in Ihre eigenen Programme einbauen können

3-89058-009-2 280 Seiten DM 49,--
komplett mit Diskette DM 89,--

In allen Buchhandlungen und Computershops oder direkt von Pandabooks, Bismarckstr. 67, 1000 Berlin 12, (030) 342 88 00

Bestellcoupon

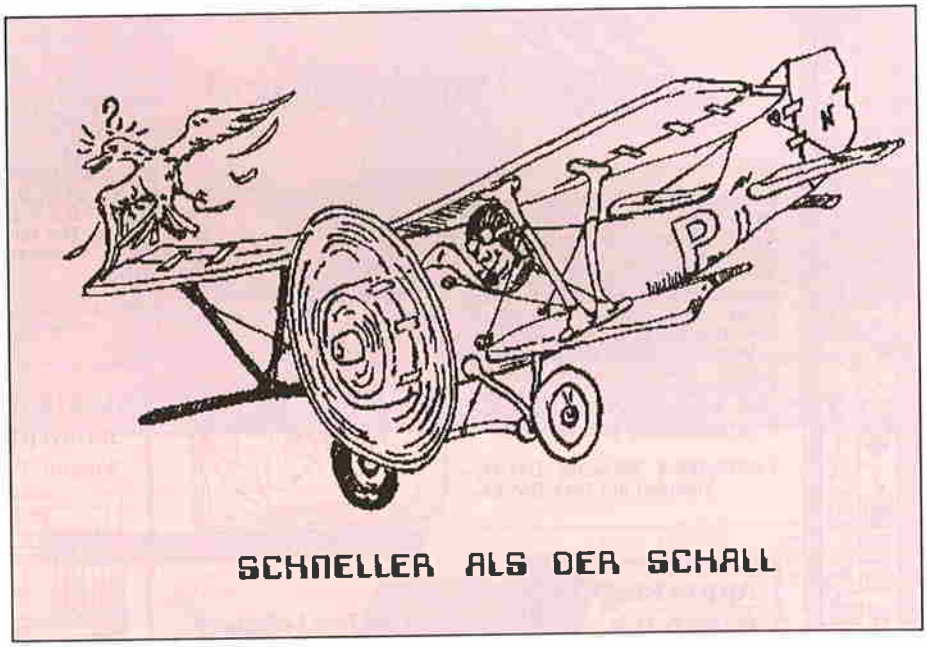
Name:
Anschrift:

O V-Scheck liegt bei (spesenfreie Lieferung)
O per Nachnahme (zzgl. Versandspesen)

Menge	Titel	Preis
1	AppleQuick	gratis

Österreich: OO Landesverlag, Linz * Schweiz: Thali AG, Hitzkirch

Einsatz kommt oder der XPLOT (8) eingeschaltet wird. SETCOL wird auch bei der folgenden Routine benutzt.



5. HINTERGRUND

Grundsätzlich funktioniert HINTERGRUND wie BOX, nur mit festen Werten und dadurch schneller (schließlich muß die ganze Seite gefüllt werden). Der entsprechende Applesoft-Befehl lautet
CALL 25177.

6. PAGE.SWAP

Für den Grafik-Editor, den ich später vorstellen werde (HGR als Arbeitsseite und HGR2 als Hilfsseite), wird es oft notwendig sein, die zwei Seiten auszutauschen. Diese kleine Routine besorgt das blitzschnell. Zum Swapppen verwende man
CALL 25258
als Applesoft-Befehl. Hier ist der Unterschied zwischen Applesoft und Maschinensprache voll spürbar. Ein entsprechendes Applesoft-Programm würde über 200mal langsamer sein!

Lassen wir es für diesmal genug sein. Im 3. Teil von Graf-quattro geht es um das Übertragen oder Kopieren von einem beliebigen viereckigen Ausschnitt, entweder auf derselben oder auch auf der anderen Grafikseite.

Der Hörer bleibt drauf...

...denn das Modem WS 2000 überträgt Daten ohne Umwege.

▶ Datenaustausch und Kommunikation – weltweit und mit praktisch jedem Computer ▶ Nutzung von Datenbanken, Mailboxen, DATEX-P, BTX, BTX rückwärts usw. ▶ TELEX nutzen ohne eigenen Anschluß – mit Ihrem Computer und dem WS 2000 ▶ Alle gängigen Baudraten (75, 300/300, 600, 1200, 1200/75, 75/1200) und internationalen Standarts (CCITT, BELL) – auch automatisch umschaltbar (mit IC-Satz SK-1) ▶ Automatisches Wählen (mit AD-2) ▶ Automatisches Anruf-Annehmen (mit AA-2) ▶ Einfacher Anschluß (parallel zur Telefonleitung) ▶ Eingebautes Netzteil ▶ Deutsche Anleitung ▶ 1 Jahr Garantie ▶ Sofort ab Lager Hamburg lieferbar

Preise (inkl. MWST.; zuzügl. Verp. u. Versandk.; bei Vorkasse frei Haus):

WS 2000	World Standard Modem	DM 798,00	IBMC	IBM-Interface (UPL-Port)	DM 330,60
AD-2	AUTODIAL-Platine	DM 199,50	APC	Apple-Int. (RS232 u. UPL)	DM 330,60
AA-2	AUTOANSWER-Platine	DM 199,50	CBM-1	Interface f. C64/VIC20	DM 136,80
SK-1	Steuer-IC-Satz	DM 96,90	ML-2	Kabel Computer/Modem	
UPL	Kabel f. AD-2 u. SK-1	DM 45,60	(nicht nötig f. APC-2 u. CBM-1)		DM 57,00

Software und weitere Interfaces auf Anfrage · Erhältlich im Fachhandel oder vom Importeur

Claus F. Erbrecht · Computer Related Products
Lappenbergsallee 37 · 2000 Hamburg 20
Tel.: 040/850 52 55

Achtung: Nur für hausinterne Telefon-Anlagen – in der Bundesrepublik Deutschland ist der Anschluß an das öffentliche Telefonnetz nicht gestattet!

MIRACLE TECHNOLOGY

CURSOR1

```
                 ORG  $60B5       ;24757
                 *
                 * CURSOR1
                 * ========
                 *
                 * von N.G. Barbieri/1985
                 *
                 * Aufrufen mit CALL 24757: X,Y
                 *
                 COLON    EQU  $3A         ;":"
                 CUXL     EQU  $3CA
                 CUXH     EQU  $3CB
                 CUY      EQU  $3CC
                 SYNCHR   EQU  $DEC0
                 HFNS     EQU  $F6B9
                 HPOSN    EQU  $F411
                 XDRAW    EQU  $F65D
                 *
                 * Weitere CALLS
                 *
                 * Hole alte Koordinate ...
                 *
60B5: AE CA 03          LDX  CUXL
60B8: AC CB 03          LDY  CUXH
60BB: AD CC 03          LDA  CUY
                 *
                 * und lösche alten Cursor.
                 *
60BE: 20 D2 60          JSR  CURS
                 *
                 * Erster CALL
                 *
                 * Hole neue Koordinate
                 *
60C1: A9 3A             LDA  #<COLON     ;Syntax-
60C3: 20 C0 DE          JSR  SYNCHR      ;prüfung
60C6: 20 B9 F6          JSR  HFNS
                 *
                 * Speichere neue Koordinate ...
                 *
60C9: 8E CA 03          STX  CUXL
60CC: 8C CB 03          STY  CUXH
60CF: 8D CC 03          STA  CUY
                 *
                 * ... und bilde neuen Cursor.
                 *
60D2: 20 11 F4   CURS   JSR  HPOSN
60D5: A2 DE             LDX  #<SHAPE     ;Shape-
60D7: A0 60             LDY  #>SHAPE     ;adresse
60D9: A9 00             LDA  #$0         ;ROT=0
60DB: 4C 5D F6          JMP  XDRAW
                 *
                 * SHAPE für Cursor +
                 *
60DE: 32 07 C1   SHAPE  HEX  3207C1
60E1: C1 67 21          HEX  C16721
60E4: 55 3A 07          HEX  553A07
60E7: 00                HEX  00
```

51 Bytes

CURSOR2

```
                 ORG  $60E8       ;24808
                 *
                 * CURSOR2
                 * ========
                 *
                 * von N.G. Barbieri/1985
                 *
                 * Aufrufen mit CALL 24808: X,Y
                 *
                 COLON    EQU  $3A         ;":"
                 CUXL     EQU  $3CD
                 CUXH     EQU  $3CE
                 CUY      EQU  $3CF
                 SYNCHR   EQU  $DEC0
                 HFNS     EQU  $F6B9
                 HPOSN    EQU  $F411
                 XDRAW    EQU  $F65D
                 *
                 * Für weitere Kommentare siehe
                 * CURSOR1!
                 *
                 * Hole alte Koordinate
                 *
60E8: AE CD 03          LDX  CUXL
60EB: AC CE 03          LDY  CUXH
60EE: AD CF 03          LDA  CUY
60F1: 20 05 61          JSR  CURS
60F4: A9 3A             LDA  #<COLON     ;Syntax-
60F6: 20 C0 DE          JSR  SYNCHR      ;check
60F9: 20 B9 F6          JSR  HFNS
                 *
                 * Speichere neue Koordinate
                 *
60FC: 8E CD 03          STX  CUXL
60FF: 8C CE 03          STY  CUXH
6102: 8D CF 03          STA  CUY
6105: 20 11 F4   CURS   JSR  HPOSN
6108: A2 11             LDX  #<SHAPE     ;Shape-
610A: A0 61             LDY  #>SHAPE     ;adresse
610C: A9 00             LDA  #$0         ;ROT=0
610E: 4C 5D F6          JMP  XDRAW
                 *
                 * SHAPE für Cursor X
                 *
6111: 1C 1C 4D   SHAPE  HEX  1C1C4D
6114: F1 1E 1E          HEX  F11E1E
6117: 1E 4D 39          HEX  1E4D39
611A: C1 07 00          HEX  C1070D
```

53 Bytes

LINIE

```
                 ORG  $611D       ;24861
                 *
                 * LINIE
                 * =====
                 *
                 * von N.G.Barbieri/1985
                 *
                 * Aufrufen mit CALL 24861
                 *
                 CU1XL    EQU  $3CA
                 CU1XH    EQU  $3CB
                 CU1Y     EQU  $3CC
                 CU2XL    EQU  $3CD
                 CU2XH    EQU  $3CE
                 CU2Y     EQU  $3CF
                 XFLAG    EQU  $300
                 CURS1    EQU  $60D2
                 CURS2    EQU  $6105
                 HPOSN    EQU  $F411
                 XLINE    EQU  $603F
                 HGLIN    EQU  $F53A
                 *
                 * Zieht eine HPLOT- oder XPLOT-
                 * Linie zwischen CURSOR1 und
                 * CURSOR2.
                 *
                 * Erst Cursoren löschen
                 *
611D: 20 46 61          JSR  CURSORE
                 *
                 * Dann Linie bilden ...
                 *
6120: AE CA 03          LDX  CU1XL
6123: AC CB 03          LDY  CU1XH
6126: AD CC 03          LDA  CU1Y
6129: 20 11 F4          JSR  HPOSN
612C: AE CE 03          LDX  CU2XH
612F: AC CF 03          LDY  CU2Y
6132: AD 00 03          LDA  XFLAG       ;Ist XOR?
6135: F0 09             BEQ  XOR
6137: AD CD 03          LDA  CU2XL       ;Nein!
613A: 20 3A F5          JSR  HGLIN
613D: 4C 46 61          JMP  CURSORE
                 *
6140: AD CD 03   XOR    LDA  CU2XL       ;Ja!
6143: 20 3F 60          JSR  XLINE
                 *
                 * ... und wieder Cursoren.
                 *
6146: AE CA 03   CURSORE LDX CU1XL
6149: AC CB 03          LDY  CU1XH
614C: AD CC 03          LDA  CU1Y
614F: 20 D2 60          JSR  CURS1
6152: AE CD 03          LDX  CU2XL
6155: AC CE 03          LDY  CU2XH
6158: AD CF 03          LDA  CU2Y
615B: 20 05 61          JSR  CURS2
615E: 60                RTS
                 *
                 * Die Routine CURSORE kann auch
```

```
                 61   * von anderen Patches
                 62   * angesprochen werden!
```

66 Bytes

VIERECK

```
                 1            ORG  $615F       ;24927
                 2   *
                 3   * VIERECK
                 4   * ======
                 5   *
                 6   * von N.G.Barbieri/1985
                 7   *
                 8   * Aufrufen mit CALL 24927
                 9   *
                 10  CU1XL    EQU  $3CA
                 11  CU1XH    EQU  $3CB
                 12  CU1Y     EQU  $3CC
                 13  CU2XL    EQU  $3CD
                 14  CU2XH    EQU  $3CE
                 15  CU2Y     EQU  $3CF
                 16  XFLAG    EQU  $3ØØ
                 17  HPOSN    EQU  $F411
                 18  XLINE    EQU  $6Ø3F
                 19  HGLIN    EQU  $F53A
                 2Ø  CURSORE  EQU  $6146
                 21  *
                 22  * Bildet ein HPLOT- oder XPLOT-
                 23  * Viereck, wobei CURSOR1 und
                 24  * CURSOR2 diagonal entgegen-
                 25  * gesetzte Ecken sind.
                 26  *
615F: 2Ø 46 61   27           JSR  CURSORE
                 28  *
6162: AE CA Ø3   29           LDX  CU1XL       ;Beginn
6165: AC CB Ø3   3Ø           LDY  CU1XH
6168: AD CC Ø3   31           LDA  CU1Y
616B: 2Ø 11 F4   32           JSR  HPOSN
                 33  *
616E: AC CC Ø3   34           LDY  CU1Y        ;Seite 1
6171: AE CE Ø3   35           LDX  CU2XH
6174: AD CD Ø3   36           LDA  CU2XL
6177: 2Ø A1 61   37           JSR  DOLINE
                 38  *
617A: AC CF Ø3   39           LDY  CU2Y        ;Seite 2
617D: AE CE Ø3   4Ø           LDX  CU2XH
618Ø: AD CD Ø3   41           LDA  CU2XL
6183: 2Ø A1 61   42           JSR  DOLINE
                 43  *
6186: AC CF Ø3   44           LDY  CU2Y        ;Seite 3
6189: AE CB Ø3   45           LDX  CU1XH
618C: AD CA Ø3   46           LDA  CU1XL
618F: 2Ø A1 61   47           JSR  DOLINE
                 48  *
6192: AC CC Ø3   49           LDY  CU1Y        ;Seite 4
6195: AE CB Ø3   5Ø           LDX  CU1XH
6198: AD CA Ø3   51           LDA  CU1XL
619B: 2Ø A1 61   52           JSR  DOLINE
                 53  *
619E: 4C 46 61   54           JMP  CURSORE
                 55  *
                 56  * Folgende Routine bildet eine
                 57  * Linie entweder in Farbe oder
                 58  * XOR. Kann auch von anderen
                 59  * Utilities angesprochen werden.
                 6Ø  *
61A1: 48         61  DOLINE   PHA
61A2: AD ØØ Ø3   62           LDA  XFLAG       ;XOR?
61A5: FØ Ø4      63           BEQ  XOR         ;Ja!
61A7: 68         64           PLA              ;Nein!
61A8: 4C 3A F5   65           JMP  HGLIN
                 66  *
61AB: 68         67  XOR      PLA
61AC: 4C 3F 6Ø   68           JMP  XLINE
```

8Ø Bytes

BOX

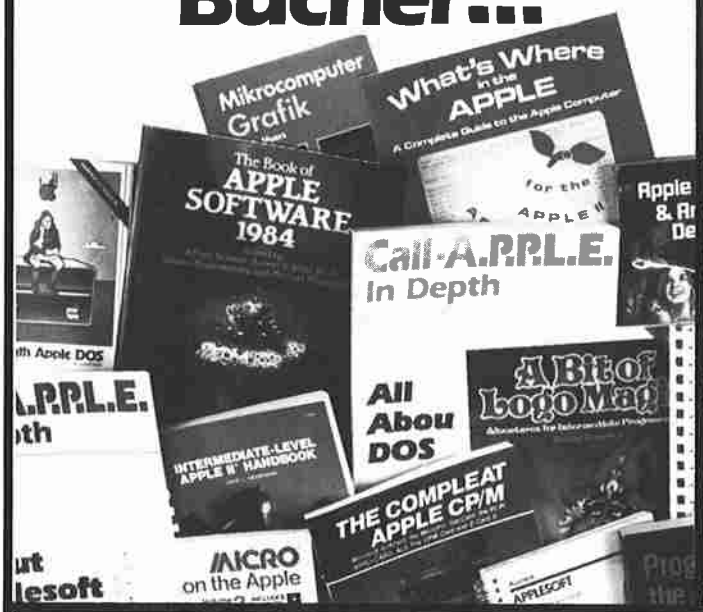
```
                 1            ORG  $61AF       ;25ØØ7
                 2   *
                 3   * BOX
                 4   * ===
                 5   *
                 6   * von N.G.Barbieri/1985
                 7   *
                 8   * Aufrufen mit CALL 25ØØ7
                 9   *
                 1Ø  CU1XL    EQU  $3CA
                 11  CU1XH    EQU  $3CB
                 12  CU1Y     EQU  $3CC
                 13  CA1Y     EQU  $3C6
                 14  CU2XL    EQU  $3CD
                 15  CU2XH    EQU  $3CE
                 16  CU2Y     EQU  $3CF
                 17  CA2Y     EQU  $3C9
                 18  XFLAG    EQU  $3ØØ
                 19  FARBE1   EQU  $3Ø2
                 2Ø  FARBE2   EQU  $3Ø3
                 21  HPOSN    EQU  $F411
                 22  HCOLORZ  EQU  $E4
                 23  CURSORE  EQU  $6146
                 24  DOLINE   EQU  $61A1
                 25  COUNT    EQU  $Ø
                 26  CFLAG    EQU  $1
                 27  COL1     EQU  $2
                 28  COL2     EQU  $3
                 29  *
                 3Ø  * Füllt eine Box alternativ
                 31  * mit zwei Farben auf, wobei
                 32  * eine oder beide Farben auch
                 33  * XOR sein können!
                 34  *
61AF: 2Ø 46 61   35           JSR  CURSORE
61B2: 2Ø 3F 62   36           JSR  SETCOL
61B5: A9 ØØ      37           LDA  #$Ø
61B7: 85 Ø1      38           STA  CFLAG
                 39  *
                 4Ø  * Die vorhandenen Y-Werte müssen
                 41  * gemerkt werden.
                 42  *
61B9: AD CC Ø3   43           LDA  CU1Y
61BC: AE CF Ø3   44           LDX  CU2Y
61BF: 8D C6 Ø3   45           STA  CA1Y
61C2: 8E C9 Ø3   46           STX  CA2Y
61C5: CD CF Ø3   47           CMP  CU2Y
                 48  *
                 49  * Wenn beide Y gleich sind,
                 5Ø  * dann keine BOX!
                 51  *
61C8: DØ 14      52           BNE  WEITER
                 53  *
                 54  * Vor RETURN sicherstellen, daß
                 55  * die ursprünglichen Y-Werte
                 56  * wieder hergestellt werden!
                 57  *
61CA: AD C6 Ø3   58  RETURN   LDA  CA1Y
61CD: AE C9 Ø3   59           LDX  CA2Y
61DØ: 8D CC Ø3   6Ø           STA  CU1Y
61D3: 8E CF Ø3   61           STX  CU2Y
61D6: A9 ØØ      62           LDA  #$Ø
61D8: 8D ØØ Ø3   63           STA  XFLAG
61DB: 4C 46 61   64           JMP  CURSORE
                 65  *
                 66  * Falls Y2>Y1, einfach umtauschen
                 67  *
61DE: BØ ØE      68  WEITER   BCS  SUBTR
61EØ: 48         69           PHA
61E1: AD CF Ø3   7Ø           LDA  CU2Y
61E4: 8D CC Ø3   71           STA  CU1Y
61E7: 68         72           PLA
61E8: 8D CF Ø3   73           STA  CU2Y
61EB: AD CC Ø3   74           LDA  CU1Y
                 75  *
                 76  * Anzahl der notwendigen Linien
                 77  * im COUNT speichern.
                 78  *
61EE: 38         79  SUBTR    SEC
61EF: ED CF Ø3   8Ø           SBC  CU2Y
61F2: 85 ØØ      81           STA  COUNT
61F4: E6 ØØ      82           INC  COUNT
                 83  *
                 84  * Siehe auch Kommentare zum
                 85  * Programm HINTERGRUND!
                 86  *
61F6: A5 Ø1      87  LOOP     LDA  CFLAG
61F8: DØ 1A      88           BNE  NEXTCOL
61FA: C6 Ø1      89           DEC  CFLAG
61FC: A5 Ø2      9Ø           LDA  COL1
61FE: C9 Ø8      91           CMP  #$8
62ØØ: DØ Ø8      92           BNE  LINCOL
                 93  *
62Ø2: A9 ØØ      94  SETXOR   LDA  #$Ø
62Ø4: 8D ØØ Ø3   95           STA  XFLAG
62Ø7: 4C 1E 62   96           JMP  LINE
                 97  *
62ØA: 85 E4      98  LINCOL   STA  HCOLORZ
```

Sie haben einen Apple...

wir haben die Software...

und die Hardware...

wir haben die Bücher...



und die Zeitschriften*...

***Fordern Sie unseren Gratiskatalog an!**

ALLES FÜR DEN APPLE II, II+, IIE

pandasoft Dr.-Ing. Eden

UHLANDSTR. 195 · D-1000 BERLIN 12

TEL.:(030) 310 423 · TELEX: 18 58 59

Autorisierter apple Fachhändler MICROSOFT Distributor

Ich besitze einen Apple. Bitte schicken Sie mir Ihren kostenlosen Katalog.

Name: _____

Adresse: _____

```
620C: A9 01      99            LDA  #$1
620E: 8D 00 03   100           STA  XFLAG
6211: 4C 1E 62   101           JMP  LINE
                 102  *
6214: E6 01      103  NEXTCOL  INC  CFLAG
6216: A5 03      104           LDA  COL2
6218: C9 08      105           CMP  #$8
621A: F0 E6      106           BEQ  SETXOR
621C: D0 EC      107           BNE  LINCOL
                 108  *
621E: AE CA 03   109  LINE     LDX  CU1XL
6221: AC CB 03   110           LDY  CU1XH
6224: AD CC 03   111           LDA  CU1Y
6227: 20 11 F4   112           JSR  HPOSN
622A: AE CE 03   113           LDX  CU2XH
622D: AC CC 03   114           LDY  CU1Y
6230: AD CD 03   115           LDA  CU2XL
6233: 20 A1 61   116           JSR  DOLINE
6236: CE CC 03   117           DEC  CU1Y
6239: C6 00      118           DEC  COUNT
                 119  *
                 120  * Noch Linien?
                 121  *
623B: F0 8D      122           BEQ  RETURN     ;Nein!
623D: D0 B7      123           BNE  LOOP       ;Ja!
                 124  *
                 125  * SETCOL sorgt dafür, daß
                 126  * in den temporären Variablen
                 127  * COL1 und COL2 zwei Farben,
                 128  * inkl. XOR, für Alternativ-
                 129  * betrieb gespeichert werden.
                 130  * Ist auch von anderen Utilities
                 131  * aus ansprechbar!
                 132  *
623F: AE 02 03   133  SETCOL   LDX  FARBE1
6242: BD 50 62   134           LDA  CTABLE,X
6245: 85 02      135           STA  COL1
6247: AE 03 03   136           LDX  FARBE2
624A: BD 50 62   137           LDA  CTABLE,X
624D: 85 03      138           STA  COL2
624F: 60         139           RTS
                 140  *
6250: 00 2A 55   141  CTABLE   HEX  002A557F80AAD5FF08
6253: 7F 80 AA D5 FF 08
```

170 Bytes

HINTERGRUND

```
                 1                ORG  $6259      ;25177
                 2    *
                 3    * HINTERGRUND
                 4    * ==========
                 5    *
                 6    * von N.G.Barbieri/1985
                 7    *
                 8    * Aufrufen mit 25177
                 9    *
                 10   XFLAG    EQU  $300
                 11   HPOSN    EQU  $F411
                 12   HCOLORZ  EQU  $E4
                 13   CURSORE  EQU  $6146
                 14   DOLINE   EQU  $61A1
                 15   SETCOL   EQU  $623F
                 16   COUNT    EQU  $0
                 17   CFLAG    EQU  $1
                 18   COL1     EQU  $2
                 19   COL2     EQU  $3
                 20   *
                 21   * Füllt mit COLOR, MIXCOLOR
                 22   * oder XORt (macht negativ)
                 23   * die HGR-Seite.
                 24   *
6259: 20 46 61   25               JSR  CURSORE
                 26   *
625C: 20 3F 62   27               JSR  SETCOL
                 28   *
625F: A9 00      29               LDA  #$0
6261: 85 01      30               STA  CFLAG
6263: A9 C0      31               LDA  #$C0
6265: 85 00      32               STA  COUNT
                 33   *
6267: C6 00      34   LOOP        DEC  COUNT
                 35   *
                 36   * CFLAG = 0  --> COLOR 1
                 37   * CFLAG <> 0 --> COLOR 2
                 38   *
6269: A5 01      39               LDA  CFLAG
626B: D0 1A      40               BNE  NEXTCOL
626D: C6 01      41               DEC  CFLAG
626F: A5 02      42               LDA  COL1
                 43   *
                 44   * Ist COLOR = XOR?
                 45   *
6271: C9 08      46               CMP  #$8
6273: D0 08      47               BNE  LINCOL     ;Nein!
                 48   *
6275: A9 00      49   SETXOR      LDA  #$0        ;Ja!
6277: 8D 00 03   50               STA  XFLAG
627A: 4C 91 62   51               JMP  LINE
                 52   *
627D: 85 E4      53   LINCOL      STA  HCOLORZ
627F: A9 01      54               LDA  #$1
6281: 8D 00 03   55               STA  XFLAG
6284: 4C 91 62   56               JMP  LINE
                 57   *
6287: E6 01      58   NEXTCOL     INC  CFLAG
6289: A5 03      59               LDA  COL2
628B: C9 08      60               CMP  #$8
628D: F0 E6      61               BEQ  SETXOR
628F: D0 EC      62               BNE  LINCOL
                 63   *
6291: A2 00      64   LINE        LDX  #$0
6293: A0 00      65               LDY  #$0
6295: A5 00      66               LDA  COUNT
6297: 20 11 F4   67               JSR  HPOSN
629A: A2 01      68               LDX  #$1
629C: A9 17      69               LDA  #$17
629E: A4 00      70               LDY  COUNT
62A0: 20 A1 61   71               JSR  DOLINE
                 72   *
                 73   * Noch Linien?
                 74   *
62A3: A5 00      75               LDA  COUNT
62A5: D0 C0      76               BNE  LOOP       ;Ja!
62A7: 4C 46 61   77               JMP  CURSORE    ;Nein!
```

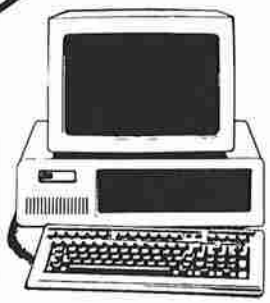
81 Bytes

PAGE.SWAP

```
                 1                ORG  $62AA      ;25258
                 2    *
                 3    * PAGE.SWAP
                 4    * ==========
                 5    *
                 6    * von N.G.Barbieri/1985
                 7    *
                 8    * Aufrufen mit CALL 25258
                 9    *
                 10   HGR      EQU  $6
                 11   HGR2     EQU  $8
                 12   PAG1     EQU  $7
                 13   PAG2     EQU  $9
                 14   *
                 15   * Diese Routine tauscht super-
                 16   * schnell, unter Verwendung aller
                 17   * 3 Register, HGR mit HGR2 aus.
                 18   *
62AA: A9 40      19               LDA  #$40
62AC: 85 09      20               STA  PAG2
62AE: A9 20      21               LDA  #$20
62B0: 85 07      22               STA  PAG1
62B2: A9 00      23               LDA  #0
62B4: 85 06      24               STA  HGR
62B6: 85 08      25               STA  HGR2
62B8: A8         26               TAY
                 27   *
62B9: B1 06      28   LOOP        LDA  (HGR),Y
62BB: AA         29               TAX
62BC: B1 08      30               LDA  (HGR2),Y
62BE: 91 06      31               STA  (HGR),Y
62C0: 8A         32               TXA
62C1: 91 08      33               STA  (HGR2),Y
62C3: C8         34               INY
                 35   *
                 36   * Ist Y-Register wieder auf 0?
                 37   *
62C4: D0 F3      38               BNE  LOOP       ;Nein!
                 39   *
62C6: E6 07      40               INC  PAG1       ;Ja!
62C8: E6 09      41               INC  PAG2
                 42   *
                 43   * Sind alle 8K übertragen
                 44   * worden?
                 45   *
62CA: A5 07      46               LDA  PAG1
62CC: C9 40      47               CMP  #$40
                 48   *
62CE: D0 E9      49               BNE  LOOP       ;Nein!
62D0: 60         50               RTS             ;Ja!
```

39 Bytes

MICROMINT



MICROMINT VOLLTREFFER

LASAR 16
IBM 264 K, TEAC B FDD, Contr. color Graphik, Multifunktionscard, Tastatur, Monitor Netzteil 15 A **4.990;**

LASAR ZE
– Apple comp. 64 K + 12 K ROM + 6502 + Z 80 A 80 Z sw Tastatur **1.290;**

Außerdem volles Rückgaberecht innerhalb 14 Tagen ohne Begründung.

	Apple	IBM
● Mehrzweckklappgehäuse lt. Abb.	179,–	179,–
● Schaltnetzteile Apple 5 A/IBM 15 A	115,–	350,–
● Profitastatur dtsch. LASAR 2000	299,–	299,–
● Interface ab	75,–	148,–
● Monitor 22 Mhz incl. Fuß, bernstein	299,–	299,–

Kaufgarantie/Tiefstpreisgarantie/1A Qualität: 100 % kompatibel inkl. Systemsoftware Made by Micromint: Apple II 495,–, Apple IIe 695,–, IBM 895,– Fertigplatinen 495/695/895. Tragbare Gehäuse für 7-Zoll/9-Zoll-Monitore 595,– DM inkl. Tastaturen. Winchester 27 MB auf Anfrage 1A First Class Controller bis 140 MB 935,– DM.

Generalimporteur MICROMINT Computer GmbH
Hochdahler Straße 151, 4006 Erkrath 2
Telex 8589305 mcm

☎ **02104/33024**

Der nächste Peeker
Heft 7/1985
erscheint am
24. 6. 1985

APPLE-INTERFACE

MESON II-PC, 48K, Apple-kompatibel . . . ab 1100,–
Z80, 16K . . . . . je 125,–
PAL, EPROM, CENTRONICS, RS 232 . . . je 170,–
80 Zeichen SoftSwitch 200,–
ADD 2B, VIA-Card (6522) mit RAM u. BackUp . 170,–
ADD 4B, PIA-Card (6821) mit RAM u. BackUp . 150,–
JOYPORT zum Anschluß von zwei Atari-Joystick 40,–
TEAC FD 55 F 580,–
Shugart SA 465 80 Track DS sehr leise . . . 550,–
r-phi-Controller AFDC4 Autopatch 300,–
LDD 103, 40 Track Apple-kompatibel Slim . . 450,–

Preise pro Stck. inkl. MwSt.

Ladenverkauf: Mo. u. Fr. 12–18 Uhr,
Mi. u. Do. 15–18 Uhr, Sa. 10–13 Uhr
Telefon 030-833 13 03

PREISH(A)EMMER

Stand 18. 4. 1985 (Tagespreise bitte telefonisch erfragen)

	1–9	10+
uPD 4116-200ns	3,30	3,–
M5K 4164 Autorafr.	11,50	11,–
TMM 41256 C15	29,90	29,–
6116 LP3 (SRM 2016)	13,–	12,–
2716-450ns	13,30	13,–
2732-450ns	16,–	15,50
2764-250ns	14,–	13,–
27128-250ns	32,–	30,–
27256-250ns	85,–	79,–
Z80 CPU	4,90	4,50
Z80A CPU	6,70	6,20
74 LS 00	–,70	–,60
74HCT 00	1,20	1,–
TDA 2030	2,75	2,50
LED 5mm Rot superhell 10 Stck.	1,80	1,50
LED 8 mm Rot	1,–	–,90

Wolfen Bauteile und Minipostaufsatz a. A. Mindestbestellwert 50,–, Versand + 10,– DM Pauschale, Versandkosten bis 250 9,–, Versand per NN oder VK, bei NN +1,50 DM

TOMBSTONE-MICRO

T. Tank & G. Körber · Gardeschützenweg 72 · 1000 Berlin 45

ZUSATZ-KARTEN:

V-24-Schnittstelle 199,– **Z-80-Karte** 139,–
80-Zeichen-Karte m.Softswitch 236,– **16 K-Language-Karte** 138,–
Centronics-Karte von Epson für Graphik . . . 210,– für Text 145,–
Centronics-Schnittstelle für 2 Drucker gleichzeitig 129,–
Eprommer incl. Software 198,–

Super-Eprommer

belegt keinen Slot, incl. Software für 2708-27128 . . . . . **239,–**
128-K-Karte . . . . . **auf Anfrage**

Floppy-Controller

FDC 4 für alle Laufwerke . . . . . 199,– Bausatz wie links 179,–
Leerplatine wie oben incl. Prom u. Eprom 130,–

Autopatch-Controller (Erphi Controller)

1 x 35 bis 2 x 80 Tr.-Disk, keine Patch-Disk notwendig, Patch DOS 3.3, Diversi-DOS 2-C, 4-C (DD MOVER), Pascal 1.1, Pascal 1.2, CPM 2.2, ProDOS 1.0.1, 1.1, 1.1.1; Sie können die Laufwerke untereinander mischen . . **298,–**

Joy Stick De Luxe . . 59,– Netzteil 5A 149,–

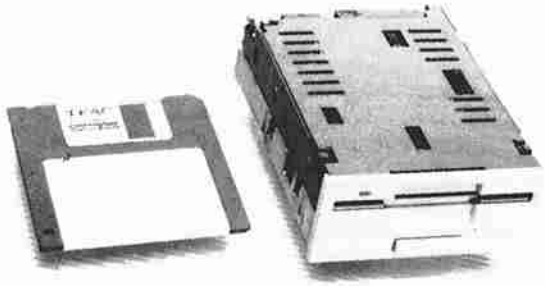
Gehäuse für 1 5¼" Slimline Laufwerk 39,–
Gehäuse für 2 5¼" Slimline Laufwerke mit Platz für ein Netzteil 159,–
Gehäuse für 2 3½" Slimline Laufwerke mit Platz für ein Netzteil 79,–
IBM®-Gehäuse 229,–
Floppy-Kabel 34pol. für 2 Laufwerke mit Shugart-Bus 35,–

Preh Commander Keyboards

Wir bieten Ihnen die **Preh-Qualität** auch für Apple. AK 88 spez. mit Gehäuse, Anschlußkabel, Zehner-Tastenfeld, dt. Zeichensatz, Sondertasten für Ctrl-Codes und Rechenfunktionen **339,–**
Preh Commander Keyboard AK 87, frei programmierbar bis zu 10 Ebenen, pro Taste bis zu 250 Zeichen . . . . . nur **559,–**

TEAC 3½" Laufwerk FD 35 F 579,–

Speicherkapazität 1 MB, (formatiert 640 KB) jetzt für nur **579,–**



Nochmalige Preissenkung bei TEAC:

TEAC FD 55 A 1 x 40 Track . . . . . 498,– **TEAC FD 55 B** 2 x 40 Track 579,–
TEAC FD 55 E 1 x 80 Track . . . . . 535,– **TEAC FD 55 F** 2 x 80 Track 598,–

Philipps Slimline Floppy X3134 A, 2x80 Track solange Vorrat . . . . . nur **569,–**
Philipps 2 x 80 Track, ⅔ Bauh. . . . . nur **459,–**
SONY 3½" Laufwerk . . . . . nur **799,–**

Sony Laufwerk für IIc . . . . . auf Anfrage

Apple®-kompatibles Laufwerk incl. Gehäuse + Kabel **599,–**

EPSON DRUCKER

EPSON FX 80 . . . . . 1670,– **EPSON FX 100** 2159,–
EPSON RX 80 . . . . . 1079,– **EPSON RX 80 FT** 1295,–

Patch-Diskette für **SONY 3½" Laufwerke** – ermöglicht die Anpassung an II/IIe und kompatible Computer **80,–**
Manual vorab 15,-DM (wird beim Kauf der Patch-Diskette angerechnet).
dto. für 5¼" Laufwerke **69,–**
Disketten Döb&Böd SS/DD 10St. **62,–** **10 Disketten** f.Sony 3½" Laufw. . **150,–**
Akustikkoppler AK 300 mit FTZ-Nr. incl. Netzteil **549,–**

Die Microfloppy mit Zukunft:

Speicherkapazität: 2 x 1 MByte formatiert: 2 x 640 kByte. Anschlußfertig mit PROM-residenter Patchsoftware für CP/M 2.2, Apple DOS 3.3, Diversi-DOS 2-C, 4-C (DD MOVER), Apple Pascal 1.1, Pascal 1.2, Pro-DOS 1.0.1, 1.1, 1.1.1 zum Preis von **1640,–**
Low Power Version 1740,–

10 MB Winchester

incl. Controler, Software und Gehäuse **4490,–**

Sonderangebot

Philipps 2 x 80 Track, ⅔ Bauh. **459,–**

Gesamt-Preisliste anfordern! **Preise inclusive gesetzlicher Mehrwertsteuer.**

UEDING electronics

Holtewiese 2
5750 Menden 1
DFÜ 02373/66877
Tel. 02373/63159

Das Farbbit

Pseudo-Double-Hires auf dem Apple II Plus

von H.-D. Siewert

Der Apple IIe mit erweiterter 80-Zeichenkarte und der IIc verfügen über eine hochauflösende Grafik von 560 * 192 Punkten. Dieser Bericht zeigt, wie man Ähnliches auch mit dem Apple II Plus und Kompatiblen erreicht.

Wie man weiß, verfügt der Apple in Farbe praktisch nur über eine Auflösung von 140 * 192 Punkten, da bei den Farben 1, 2, 5 und 6 nur jeder zweite Punkt gezeichnet wird. Ich selbst verzichtete daher in meinen früheren Programmen meist auf Farben, da ich zum einen auf die volle Auflösung Wert lege und zum anderen nicht über einen Farbbildschirm verfüge – wie wohl sehr viele andere auch.

Neuerdings experimentiere ich damit, dem Apple eine höhere Auflösung zu verschaffen, indem ich das Farbbit gezielt einsetze, was auf Grün-Monitoren eine etwas andere Wirkung als auf Farbbildschirmen hat. Zum Einstieg schlage ich Ihnen vor, zunächst das Programm **WANDERNDER-.STRICH** einzugeben. Sie sehen nach dem Start zwei Striche, die sich von links nach rechts über den Bildschirm bewegen. Dabei macht der obere 560, der untere aber nur 280 Schritte. Wie ist dies nun zu erklären?

Es sei zunächst die Art beschrieben, wie der Apple seine Grafik abspeichert. Pro Zeile sind wirklich nur 280 Punkte vorhanden, egal was Sie jetzt auch denken mögen. Diese Punkte werden in 40 Bytes abgespeichert, wobei jedes Byte 7 Punkte beinhaltet. Es bleibt also ein Bit übrig, welches für die farbliche Darstellung von Bedeutung ist. Es wird daher das „Farbbit" genannt. Die sieben Punkte sind nun von links nach rechts im Byte gespeichert, wobei man sich vorstellen muß, daß das ganz links stehende das niederwertigste Bit ist (das linke Bit steht also für den rechten Punkt der 7-Punkte-Gruppe usw.). Das Farbbit ist das höchstwertige Bit; es befindet sich, wie aus **Abb. 1** ersichtlich, ganz rechts. Wie die einzelnen Zeilen im Speicher stehen, interessiert in diesem Zusammenhang nicht; dazu nur soviel: Sie sind aus hardwaretechnischen Gründen nicht linear angeordnet, sondern ähnlich wie die Textzeilen nach einem bestimmten Muster gegliedert.

Das Farbbit legt nun den Farbmodus für die 7 in diesem Byte gespeicherten Punkte fest. Man kann also die Farbe nur „gebietsweise" definieren, was ganz besonders zum Tragen kommt, wenn man einen grünen Strich durch eine orangefarbene Fläche zeichnen läßt. Bei den Farben 0 bis 3 wird das Farbbit jeweils auf 0 gesetzt, bei den Farben 4 bis 7 jeweils auf 1. Auf grünen Monitoren bewirkt das Farbbit nun keine Farbveränderung, sondern eine Verschiebung um einen halben Schritt nach rechts, was auf Farbbildschirmen den Effekt hat, daß die Punkte auf anderen Farben des Farbrasters zu liegen kommen.

Mittels dieses Farbbits kann man nun eine pseudo-doppelthochauflösende Grafik auch für die älteren Apple-Versionen schaffen. Es gibt dabei zwei Methoden, mit deren Hilfe stehende Bilder höher aufgelöst erscheinen:

1 Byte des Grafikspeichers :

Bit :	0	1	2	3	4	5	6	7
Wert :	2^0	2^1	2^2	2^3	2^4	2^5	2^6	2^7
Bedeutung :	Punkt 1	Punkt 2	Punkt 3	Punkt 4	Punkt 5	Punkt 6	Punkt 7	Farbbit

Abbildung 1

Die erste Methode

Methode 1 besteht darin, daß man bei den Punkten, die eigentlich bei einem Kreis oder einem Strich zwischen den normalen Punkten stehen müßten, das Farbbit auf 1 setzt, wodurch dieser Punkt um einen halben Schritt nach rechts verschoben wird. Ich habe diese Technik bei dem Programm **KREIS.1** angewandt, bei dem das Farbbit mit Hilfe von „HCOLOR = 3 + 4 * (S - .5 > INT(S))" einmal gesetzt wird und einmal nicht. Dabei ist der logische Ausdruck „(S -.5 > (INT (S))" gleich 1, wenn hinter dem Komma bei S ein Wert steht, der größer als 0,5 ist. Das Programm zeichnet zwei Kreise, von denen der eine normal und der andere höher aufgelöst ist. Diese Methode hat natürlich ihre Vor- und Nachteile. Vorteile: Das Bild erscheint höher aufgelöst und steht ruhig. Nachteile: Es sind nur 280 Punkte pro Zeile möglich und man kann die Verschiebung der Punkte genau wie die Farbe nur 7-punkteweise bestimmen.

Die zweite Methode

Methode 2 besteht darin, daß man die normalen Punkte auf die eine Grafikseite und die verschobenen Punkte auf die andere Seite zeichnet. Um beide Bilder zusammenzubringen, muß ständig zwischen den beiden Seiten hin- und hergeschaltet werden. Das Programm für dieses Umschalten ist aus Geschwindigkeitsgründen in Assembler geschrieben und heißt **FLIP-**

PER (abzuspeichern mit „BSAVE FLIPPER, A$300 ,L$30"). Das Applesoft-Programm, das die entsprechenden Grafiken zeichnet (wieder zwei Kreise), heißt **KREIS.2**. **KREIS.3** zeichnet 3 Kreise, und zwar den linken nach Methode 2, den mittleren mit normaler Auflösung und den rechten nach Methode 1. Methode 2 hat den Vorteil, daß wirklich 560 Punkte dargestellt werden können, nachteilig ist jedoch, daß das Bild flimmert.

Es hängt vom Einzelfall ab, welche Methode man verwendet, da keine allgemeingültigen Regeln aufgestellt werden können. Zu sagen bleibt, daß sich manche Bilder mit keiner der beiden von mir vorgestellten Möglichkeiten darstellen lassen. Dazu gehören vor allen Dingen Bilder, bei denen zu viele feine Einzelheiten zu sehen sind. Will man also eine echte Auflösung von 560 * 192 Punkten, so ist doch ein Apple IIe mit erweiterter 80-Zeichenkarte oder ein IIc vorzuziehen.

Damit Sie aber auch die pseudo-doppelthochauflösende Grafik ausprobieren können, habe ich das Programm **KOMPRESSOR** geschrieben, das mit „BSAVE KOMPRESSOR, A$8000, L$E9" gespeichert werden soll. Dieses Programm faßt die Bilder beider Grafikseiten auf einer einzigen Seite zusammen. Die genauere Arbeitsweise ist aus **Abb. 2** und **3** zu ersehen. Es werden die Punkte der beiden Seiten abwechselnd auf Seite 1 und auf Seite 2 gespeichert. Dabei ist auf Seite 1 das Farbbit stets auf 0, auf Seite 2 hingegen immer auf 1 gesetzt. Wenn man nun mit FLIPPER zwischen den beiden Seiten hin- und herschaltet, wirkt das Bild doppelthochaufgelöst. Wie man KOMPRESSOR anwenden kann, können Sie aus dem Programm **KOMPRESSOR.DEMO** ersehen. Will man ein hochaufgelöstes Bild erstellen, so zeichnet man auf Seite 1 die eine und auf Seite 2 die andere Hälfte des Bildes, läßt den KOMPRESSOR seine Arbeit verrichten und startet dann FLIPPER.

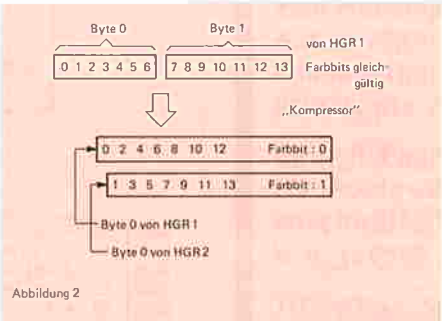


Abbildung 2

Zusammenfassend ist zu sagen, daß man durch meine Methoden nur in Ausnahmefällen etwas Ähnliches wie eine doppelthochauflösende Grafik erzeugen kann. Es gibt aber noch andere Anwendungen für das Farbbit, die nicht mit Farbe, sondern mehr mit Auflösung zu tun haben – im Bereich der Animation. Objekte, die sich waagerecht über den Bildschirm bewegen, wie z.B. der wandernde Strich, können so wesentlich feinere Schritte ausführen.

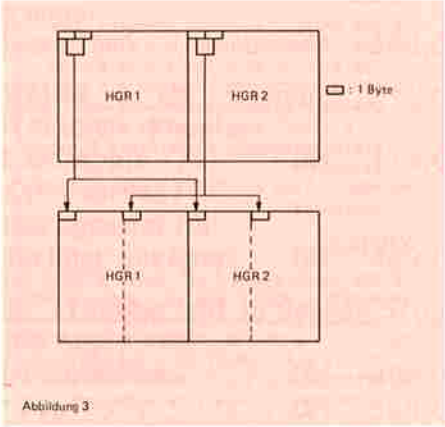


Abbildung 3

Ich hoffe, daß möglichst viele Leser durch diesen Artikel zum Experimentieren angeregt wurden, und wünsche allen dabei viel Erfolg.

Apple und IBM kompatible Computer

16K, Z80, Diskcontroler je 110,–
80 Zeichenkarte mit Softswitch
2 Zeichensätze 195,–
Motherboard 48K ohne
Firmware 610,–
Erphi-controler mit Autopatch 300,–
Siemenslaufwerk F 122 515,–
Philips X3134 2×80 Track 605,–
TEAC FD-55B 2×40 Track 515,–
TEAC FD-55F 2×80 Track 610,–
Megaboard (Mainboard XT) 1695,–
Neu: Stardrucker SG 10 920,–
Drive Controler 395,–
Monochrome Monitore ..... ab 375,–
Farbmonitore ..... ab 998,–
Tastaturen für IBM und Apple ..... ab 330,–

Versand nur per Nachnahme oder Vorkasse

Weiteres Zubehör für Apple und IBM gegen frankierten Rückumschlag.

Preissenkung:
Saturn 128K Karte 395,–

Ulf Mohwinkel Electronic
Berliner Straße 73
5090 Leverkusen Fettehenne
Telefon 02 14/9 37 81

Messen, Steuern Regeln
mit APPLE oder Kompatiblen

Peripheriekarten für technische Anwendungen:
+ Parallelports
+ A/D-Wandler
+ RS 232C-Schnittstelle
+ Real/Time Clock
außerdem:
+ Opto. Ein/Ausgabekarten

Information und Händleranfragen bei:

Entwicklungsbüro Brecht
Uhlandstraße 70
7012 Fellbach-Schmiden
Telefon (0711) 51 41 60

Preisliste kostenlos!

Unser Angebot im Juni

Drucker Star SG-10

schönschreib- und grafikfähig DM 898,–
grafikfähiges Interface hierzu DM 118,–

D.O.S. Computersysteme
Am Kühnbach 42, 7170 Schwäbisch Hall 11
Telefon (07 91) 5 17 36

WANDERNDER.STRICH

```
10 REM WANDERNDER.STRICH von H.-D. Siewert, Dez. 1984
20 HGR : NORMAL : HOME :Y1 = 80:Y2 = 100:B = 25: VTAB 21:
   PRINT "  DER OBERE STRICH MACHT 560 SCHRITTE!"
30 FOR X = 0 TO 279: HCOLOR= 3: HPLOT X,Y1 TO X,Y1 + 10:
   HPLOT X,Y2 TO X,Y2 + 10: REM Strich 1 und 2 zeichnen
40 FOR A = 0 TO B: NEXT : HCOLOR= 7: HPLOT X,Y1 TO X,Y1 +
   10: REM Strich 1 einen halben Schritt weiterrücken
50 FOR A = 0 TO B: NEXT : HCOLOR= 0: HPLOT X,Y1 TO X,Y1 +
   10: HPLOT X,Y2 TO X,Y2 + 10: REM Beide Striche
   wieder löschen
60 NEXT : TEXT
```

KREIS.1

```
10 REM KREIS.1 von H.-D. Siewert, Dez. 1984
20 HGR : NORMAL : HOME :O = 0:PI = 3.1415926535:P = PI /
   180:X1 = 93:X2 = 186:Y = 90:R = 55
30 FOR Z = O TO 360: HCOLOR= 3:S = R * SIN (Z * P):C = R *
   COS (Z * P): HPLOT X2 + S,Y + C: HCOLOR= 3 + 4 * (S
   - .5 > INT (S)): HPLOT X1 + S,Y + C: NEXT : REM Zwei
   Kreise zeichnen, davon Kreis 2 normal, Kreis 1 nach
   Methode 1
40 HOME : VTAB 21: PRINT " Höhere Auflösung    Normale
   Auflösung"
```

KREIS.2

```
10 REM KREIS.2 von H.-D. Siewert, Dez. 1984
20 NORMAL : HOME : HGR :O = 0:P = 3.1415926535 / 180:X1 =
   93:X2 = 186:Y = 90:R = 50
30 FOR Z = O TO 360: HCOLOR= 3:S = R * SIN (Z * P):C = R *
   COS (Z * P): HPLOT X1 + S + .5,Y + C: HCOLOR= 7:
   HPLOT X2 + S,Y + C: NEXT : REM Zwei Kreise auf Seite
   1 zeichnen, Kreis 1 nach Methode 2, Kreis 2 normal
40 HGR2 : FOR Z = O TO 360:S = R * SIN (Z * P):C = R * COS
   (Z * P): HPLOT X1 + S,Y + C: HPLOT X2 + S,Y + C:
   NEXT : REM Beide Kreise auf Seite 2 zeichnen, Kreis
   2 wie oben, Kreis 1 gegenüber Kreis 1 auf Seite 1
   modifiziert
50 PRINT CHR$ (4)"BRUN FLIPPER": REM FLIPPER starten
60 GET XY$: IF XY$ < > CHR$ (27) THEN CALL 768: GOTO 60:
   REM Mit ESC kann aus den Programm ausgestiegen
   werden, wenn das Bild steht
70 TEXT
```

KREIS.3

```
10 REM KREIS.3 von H.-D. Siewert, Dez. 1984
20 TEXT : NORMAL : HOME : HGR :O = 0:P = 3.1415926535 /
   180:F = .5:X1 = 70:X2 = 140:X3 = 210:Y = 90:R = 50
30 FOR Z = O TO 360: HCOLOR= 3:S = R * SIN (Z * P):C = R *
   COS (Z * P): REM Sinus- und Cosinuswerte berechnen
35 HPLOT X1 + S + F,Y + C: HCOLOR= 7: HPLOT X2 + S,Y + C:
   HCOLOR= 3 + 4 * (S - F > INT (S)): HPLOT X3 + S,Y +
   C: NEXT : REM Alle drei Kreise auf Seite 1 zeichnen,
   Kreis 1 nach Methode 2, Kreis 2 normal, Kreis 3 nach
   Methode 1
40 HGR2 : FOR Z = O TO 360: HCOLOR= 7:S = R * SIN (Z *
   P):C = R * COS (Z * P): REM Sinus- und Cosinuswerte
   berechnen
45 HPLOT X1 + S,Y + C: HPLOT X2 + S,Y + C: HCOLOR= 3 + 4 *
   (S - F > INT (S)): HPLOT X3 + S,Y + C: NEXT : REM
   Alle drei Kreise auf Seite 2 zeichnen, Kreis 2 und 3
   wie auf Seite 1, Kreis 3 modifiziert
50 PRINT CHR$ (4)"BRUN FLIPPER": REM FLIPPER starten
60 GET XY$: IF XY$ < > CHR$ (27) THEN CALL 768: GOTO 60:
   REM Wenn das Bild steht (FLIPPER außer Betrieb),
   kann man mit ESC das Programm verlassen
70 TEXT
```

KOMPRESSOR.DEMO

```
10 REM KOMPRESSOR.DEMO von H.-D. Siewert, Dez. 1984
20 NORMAL : HOME : HGR2 : HCOLOR= 7: HPLOT 279,0 TO 0,96
   TO 279,191: HPLOT 0,0 TO 279,191: HPLOT 279,0 TO
   0,191
25 REM In Zeile 20 wird die eine Hälfte des Bildes auf
   HGR2 gezeichnet.
30 HGR : POKE - 16302,0: HPLOT 0,0 TO 279,96 TO 0,191:
   HPLOT 0,0 TO 279,191: HPLOT 279,0 TO 0,191
35 REM Die andere Hälfte des Bildes wird auf HGR
   gezeichnet.
40 PRINT CHR$ (4)"BRUN KOMPRESSOR": PRINT CHR$ (4)"BRUN
   FLIPPER"
50 GET XY$: REM Nachdem das Bild gezeichnet wurde, werden
   KOMPRESSOR und FLIPPER gestartet.
60 HGR2 : HCOLOR= 7: HPLOT 0,0: CALL 62454: HCOLOR= 0:
   HPLOT 279,0 TO 0,96 TO 279,191: HPLOT 0,0 TO
   279,191: HPLOT 279,0 TO 0,191
65 REM Auf HGR2 wird ein neues Bild gezeichnet.
70 HGR : POKE - 16302,0: HCOLOR= 7: HPLOT 0,0: CALL 62454:
   HCOLOR= 0: HPLOT 0,0 TO 279,96 TO 0,191: HPLOT 0,0
   TO 279,191: HPLOT 279,0 TO 0,191
75 REM auf HGR wieder die andere Hälfte.
80 CALL 32768: CALL 768: REM Es werden erneut KOMPRESSOR
   und FLIPPER gestartet - diesmal direkt mit CALL.
```

FLIPPER

```
                1   ********************************
                2   *                              *
                3   *            Flipper           *
                4   *                              *
                5   *   Dieses Programm schaltet   *
                6   * ständig zwischen fullscreen- *
                7   * GRAFIK PAGE 1 und 2 hin und  *
                8   * her. Von H.-D. Siewert,      *
                9   *       Dezember 1984          *
                10  *                              *
                11  ********************************
                12          ORG  $300
                13  NOMIX   EQU  $C052
                14  HIRES   EQU  $C057
                15  GRAPHIC EQU  $C050
                16  PAGE1   EQU  $C054
                17  PAGE2   EQU  $C055
                18  KBD     EQU  $C000
                19  STROBE  EQU  $C010
                20  DUMMY   EQU  $00
                21
0300: AD 52 C0  22          LDA  NOMIX
0303: AD 57 C0  23          LDA  HIRES
0306: AD 50 C0  24          LDA  GRAPHIC
0309: AD 54 C0  25  LOOP    LDA  PAGE1
030C: 20 1E 03  26          JSR  WAITLONG
030F: AD 55 C0  27          LDA  PAGE2
0312: 20 20 03  28          JSR  WAIT
0315: 2C 00 C0  29          BIT  KBD
0318: 10 EF     30          BPL  LOOP
031A: 2C 10 C0  31          BIT  STROBE
031D: 60        32          RTS
                33  * Verzögerung durch LDA DUMMY
031E: A5 00     34  WAITLONG LDA DUMMY
0320: A0 0F     35  WAIT    LDY  #$0F
0322: A2 00     36  LOOP1   LDX  #$00
0324: E8        37  LOOP2   INX
0325: D0 FD     38          BNE  LOOP2
0327: 88        39          DEY
0328: D0 F8     40          BNE  LOOP1
032A: A2 C1     41          LDX  #$C1
032C: CA        42  LOOP3   DEX
032D: D0 FD     43          BNE  LOOP3
032F: 60        44          RTS
```

48 Bytes

HINWEIS: Die Zeitverzögerung der Warteschleife kann sehr genau abgeglichen werden. Die hier angegebenen Werte wurden empirisch ermittelte, um ein maximal flimmerfreies Bild zu erhalten.

KOMPRESSOR

```
1   ********************************
2   *                              *
3   *          KOMPRESSOR          *
4   *                              *
5   * Dieses Programm faßt unter   *
6   * Ausnutzung des Farbbits      *
7   * beide Grafik-Seiten so zu-   *
8   * sammen, daß sie nebeneinan-  *
9   * der dargestellt werden kön-  *
10  * nen                          *
11  * Von H.-D. Siewert, Dez. 84   *
12  *                              *
13  ********************************
14
15          ORG  $8000
16
17  YPOS    EQU  $06
18  C       EQU  $CE
19  D       EQU  $CF
20  LBAS    EQU  $FA
21  MBAS    EQU  $FC
22  HBAS    EQU  $FE
```

Sie müssen Kalkulationen erstellen ...

Sie müssen Rechnungen schreiben...

Sie müssen schnell ein Angebot zusammenstellen...

Sie brauchen eine zuverlässige Lagerverwaltung

Sie wollen Geschäftsbriefe schreiben

Dann brauchen Sie

HandMac

Das Programm für das Handwerk.

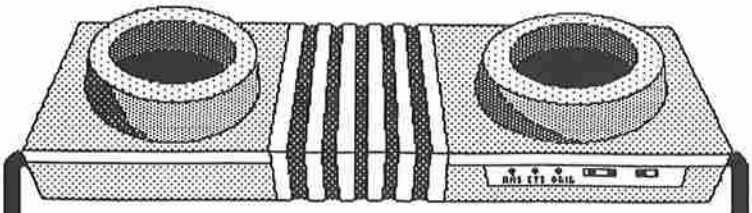
- das Ihnen diese Arbeiten abnimmt
- das speziell für Sie in Deutschland entwickelt wurde
- das Sie sofort benutzen können, ohne mehrwöchige Schulung, ohne seitenstarkes Handbuch

HandMac

läuft auf Apple Macintosh und gibt es nur im autorisierten Fachhandel

Information: copy team gmbh
schuhstr. 23
8520 erlangen
09131 – 21383

Apple und Macintosh sind eingetragene Warenzeichen der Apple Computer Inc.



AKUSTIK-KOPPLER **Dataphon s21d**

300 Baud Modem, nach CCITT V.21 Standard, mit FTZ-Nr. 18.13.1917.00, Gebühren- und anmeldefrei, V24/RS-232 Standard-Schnittstelle (25-Pin), Vollduplexbetrieb, Answer-, Originate- und Auto-Modus nur DM 298,00

TELEKOMMUNIKATIONS - KOMPLETT - PAKET

geeignet für Apple //+ und Apple //e:
1 Dataphon s21d,
1 Anschlußkabel (RS-232-Schnittstelle zum Apple II-Gameport),
1 Terminalprogramm: "HIB Modem-Transfer" nur DM 398,00

TELEKOMMUNIKATION am Apple //c:

Anschlußkabel (Akustik-Koppler zum Modemport am Apple //c) DM 114,00
ASCII-PRO (ASCII-Express) Kommunikationsprogramm für alle Apple //-Rechner geeignet DM 489,00

Chinon-Laufwerk (Testbericht in Peeker 5/85)

für Apple //+ und Apple //e
anschlußfertig im Gehäuse DM 498,00
w.o. jedoch für Apple //c DM 569,00

TOSHIBA Spitzenlaufwerke zum Superpreis!

ND 06-D, 2 x 80 Track DM 549,00
ND 04-D, 2 x 40 Track DM 498,00

DISK-DOPPEL-STATION (APPLE //+, APPLE //e)

2 x ND 06-D im Gehäuse +Auto-Patch-Controller DM 1698,00

AUTO-PATCH-CONTROLLER DM 298,00

BROTHER-Matrixdrucker, die Super-Drucker!

M-1009 (Matrixdrucker RS-232 + Centronics) DM 698,00
M-1009 anschlußfertig an:
Apple //c (mit Kabel) DM 798,00
Apple //e (mit Graphik-Interface und Kabel) DM 898,00

HR-15 XL, Typenraddrucker der Spitzenklasse!

mit Centronics-Interface DM 1698,00
HR-15 XL anschlußfertig an:
Apple //c DM 1898,00
Apple //e DM 1998,00
Apple Macintosh DM 1998,00

HR-10, der neue Low - Cost - Typenraddrucker

für alle Rechner geeignet DM 998,00

Alle Preise inclusive der gesetzlichen Mehrwertsteuer.
Berechnung der Versandkosten erfolgt nach Entfernung und Gewicht.
Fordern Sie noch heute unsere Gratispreisliste an! Wiederverkäufer bitte nur schriftlich anfragen (Kopie der Gewerbeanmeldung beilegen!).

hib

HIB Computerladen
Äußere Bayreuther Str. 72
Postfach 21 01 25
8500 Nürnberg 21
Telefon: 0911 / 515 939
Telex 17 - 911 8253

```
                23  *
                24  * Als erstes wird das Bild
                25  * in Seite 1 aufgeteilt
                26  * und die eine Hälfte in
                27  * Seite 1 abgespeichert,
                28  * die andere Hälfte in Seite 3.
                29  * Seite 3 beginnt bei $6000
                30  * und kann ohne Hardware-
                31  * änderungen nicht darge-
                32  * stellt werden.
                33  *
8000: D8        34          CLD
8001: A9 00     35          LDA  #$00
8003: 85 06     36  NEXTY1  STA  YPOS
8005: A0 00     37          LDY  #$00
8007: 20 A8 80  38          JSR  CALC
800A: B1 FA     39  NEXTX1  LDA  (LBAS),Y
800C: 20 78 80  40          JSR  CONVERTA
800F: C8        41          INY
8010: B1 FA     42          LDA  (LBAS),Y
8012: 20 8E 80  43          JSR  CONVERTB
8015: 98        44          TYA
8016: 4A        45          LSR
8017: 20 DA 80  46          JSR  STORE
801A: 0A        47          ASL
801B: A8        48          TAY
801C: C8        49          INY
801D: C8        50          INY
801E: C0 28     51          CPY  #$28
8020: D0 E8     52          BNE  NEXTX1
8022: A5 06     53          LDA  YPOS
8024: 69 00     54          ADC  #$00
8026: C9 C0     55          CMP  #$C0
8028: D0 D9     56          BNE  NEXTY1
                57  *
                58  * Hier wird dasselbe wie
                59  * oben für Seite 2 durch-
                60  * geführt. Dabei wird wieder
                61  * die eine Hälfte in Seite 1
                62  * und die andere in Seite 3
                63  * gespeichert; diesmal aber
                64  * in der rechten Hälfte der
                65  * beiden Seiten.
                66  *
802A: A9 00     67          LDA  #$00
802C: 85 06     68  NEXTY2  STA  YPOS
802E: A0 00     69          LDY  #$00
8030: 20 A8 80  70          JSR  CALC
8033: B1 FC     71  NEXTX2  LDA  (MBAS),Y
8035: 20 78 80  72          JSR  CONVERTA
8038: C8        73          INY
8039: B1 FC     74          LDA  (MBAS),Y
803B: 20 8E 80  75          JSR  CONVERTB
803E: 98        76          TYA
803F: 4A        77          LSR
8040: 18        78          CLC
8041: 69 14     79          ADC  #$14
8043: 20 DA 80  80          JSR  STORE
8046: 38        81          SEC
8047: E9 14     82          SBC  #$14
8049: 0A        83          ASL
804A: A8        84          TAY
804B: C8        85          INY
804C: C8        86          INY
804D: C0 28     87          CPY  #$28
804F: D0 E2     88          BNE  NEXTX2
8051: A5 06     89          LDA  YPOS
8053: 69 00     90          ADC  #$00
8055: C9 C0     91          CMP  #$C0
8057: D0 D3     92          BNE  NEXTY2
8059: 18        93          CLC
                95  * Hier wird als letztes das
                96  * Bild in Seite 3 nach Seite 2
                97  * bewegt, wo es das alte Bild
                98  * verdeckt.
805A: A2 40     100         LDX  #$40
805C: A9 60     101         LDA  #$60
805E: 8D 68 80  102         STA  LOOP+2
8061: A0 00     103 MOVE    LDY  #$00
8063: 8E 6B 80  104         STX  LOOP+5
8066: B9 00 60  105 LOOP    LDA  $6000,Y
8069: 99 00 40  106         STA  $4000,Y
806C: C8        107         INY
806D: D0 F7     108         BNE  LOOP
806F: EE 68 80  109         INC  LOOP+2
8072: E8        110         INX
8073: E0 60     111         CPX  #$60
8075: D0 EA     112         BNE  MOVE
8077: 60        113         RTS
                115 * Unterprogramme :
                117 * CONVERTA gliedert das erste
                118 * Byte in zwei neue Bytes auf.
                119 * Mit CONVERTB wird das zweite
                120 * Byte aufgegliedert und füllt
                121 * die beiden neuen Bytes auf.
                122 * CALC berechnet die Anfangs-
                123 * adresse der jeweiligen im
                124 * Akkumulator gespeicherten
                125 * Grafikzeile.
                126 * STORE speichert die neuen
                127 * mit CONVERTA und CONVERTB
                128 * gebildeten Bytes ab.
8078: 4A        130 CONVERTA LSR
8079: 66 CE     131         ROR  C
807B: 4A        132         LSR
807C: 66 CF     133         ROR  D
807E: 4A        134         LSR
807F: 66 CE     135         ROR  C
8081: 4A        136         LSR
8082: 66 CF     137         ROR  D
8084: 4A        138         LSR
8085: 66 CE     139         ROR  C
8087: 4A        140         LSR
8088: 66 CF     141         ROR  D
808A: 4A        142         LSR
808B: 66 CE     143         ROR  C
808D: 60        144         RTS
808E: 4A        145 CONVERTB LSR
808F: 66 CF     146         ROR  D
8091: 4A        147         LSR
8092: 66 CE     148         ROR  C
8094: 4A        149         LSR
8095: 66 CF     150         ROR  D
8097: 4A        151         LSR
8098: 66 CE     152         ROR  C
809A: 4A        153         LSR
809B: 66 CF     154         ROR  D
809D: 4A        155         LSR
809E: 66 CE     156         ROR  C
80A0: 4A        157         LSR
80A1: 66 CF     158         ROR  D
80A3: 66 CF     159         ROR  D
80A5: 66 CE     160         ROR  C
80A7: 60        161         RTS
80A8: AA        162 CALC    TAX
80A9: 29 C0     163         AND  #$C0
80AB: 85 FA     164         STA  LBAS
80AD: 4A        165         LSR
80AE: 4A        166         LSR
80AF: 05 FA     167         ORA  LBAS
80B1: 85 FA     168         STA  LBAS
80B3: 8A        169         TXA
80B4: 85 FB     170         STA  LBAS+1
80B6: 0A        171         ASL
80B7: 0A        172         ASL
80B8: 0A        173         ASL
80B9: 26 FB     174         ROL  LBAS+1
80BB: 0A        175         ASL
80BC: 26 FB     176         ROL  LBAS+1
80BE: 0A        177         ASL
80BF: 66 FA     178         ROR  LBAS
80C1: A5 FB     179         LDA  LBAS+1
80C3: 29 1F     180         AND  #$1F
80C5: AA        181         TAX
80C6: 09 20     182         ORA  #$20
80C8: 85 FB     183         STA  LBAS+1
80CA: 8A        184         TXA
80CB: 09 40     185         ORA  #$40
80CD: 85 FD     186         STA  MBAS+1
80CF: 09 20     187         ORA  #$20
80D1: 85 FF     188         STA  HBAS+1
80D3: A5 FA     189         LDA  LBAS
80D5: 85 FC     190         STA  MBAS
80D7: 85 FE     191         STA  HBAS
80D9: 60        192         RTS
80DA: A8        193 STORE   TAY
80DB: A5 CF     194         LDA  D
80DD: 09 80     195         ORA  #$80
80DF: 91 FE     196         STA  (HBAS),Y
80E1: A5 CE     197         LDA  C
80E3: 29 7F     198         AND  #$7F
80E5: 91 FA     199         STA  (LBAS),Y
80E7: 98        200         TYA
80E8: 60        201         RTS
233 Bytes
```

Peeker-Sammeldisketten

Einzelbezug DM 28,–
Fortsetzungsbezug DM 20,–
(Jederzeit kündbar, jedoch mindestens 6 Disketten)
(* = nur auf Diskette, nicht im Peeker gelistet! Seitenangaben beziehen sich auf Beginn des Listings)
Hüthig Software Service
Postfach 10 28 69 · 6900 Heidelberg 1

Disk# 1
(Heft 1+2, 1984)

T.DISASSEMBLER.65C02 (1/84, S. 15)
DISASSEMBLER.65C02

T.ACCEL.WAIT (1/84, S. 22)
ACCEL.WAIT
T.ACCEL.BOOT
ACCEL.BOOT
ACCEL.LC.KOPIERER
T.ACCEL.LC.KOPIE
ACCEL.LC.KOPIE
T.ACCEL.ROM.KOPIE1
ACCEL.ROM.KOPIE1
T.ACCEL.ROM.KOPIE2
ACCEL.ROM.KOPIE2

TURTLE.GRAFIK.MIT.REMS (1/84, S.29)
TURTLE.GRAFIK.OHNE.REMS *

DOUBLE.LORES.SOFTSWITCH.DEMO (1/84, S. 37)
DOUBLE.LORES.APPLESOFT.DEMO
AMPER.DOUBLE.LORES.DEMO
T.AMPER.DOUBLE.LORES
AMPER.DOUBLE.LORES
T.DOUBLE.LORES
DOUBLE.LORES

HIRES (1/84, S. 41)
T.PRINTHIRES
PRINTHIRES

DHGR.APSOFT.DEMO (2/84, S. 30)
AMPER.DOUBLE.HIRES.BAS
AMPER.DOUBLE.HIRES
T.AMPER.DOUBLE.HIRES
DHGR.LINEPLOTTER

INSTRING.TEST (2/84, S. 43)
INSTRING.OBJ
T.INSTRING.OBJ
INSTRING.LISA.SOURCE

LOESCHEN.EINES.ARRAYS (2/84, S. 52)

ULTRATERM.ENGLISCH * (2/84, S. 60)
ULTRATERM.DEUTSCH *

PRIMZAHLEN.OVERMEYER * (2/84, S. 70)
PRIM.OBJ0 *
PRIM.OBJ1 *
PRIM.TEST *
PRIM.TOOLKIT.SOURCE *

Disk #2
(Heft 1-2, 1985, DOS-Format)

T.RAMDISKLC (1-2/85, S. 14)
RAMDISKLC

T.IBS.RAMDISKDRIVER (1-2/85, S. 20)
IBS.RAMDISKDRIVER
T.AP20.RAMDISKTEST
AP20.RAMDISKTEST

T.QUICKCOPY (1-2/85, S. 26)
QUICKCOPY
QUICKCOPY.PUFFER
PRODOS.COPYA
T.PRODOS.COPYOBJ *
PRODOS.COPYOBJ

PRODOS.PATCH (1-2/85, S. 31)

T.APPLESOFT.FRE (1-2/85, S. 36)
T.LC.FRE
LC.FRE
FRE.TEST
T.RAM.FRE *
RAM.FRE

T.SCHIRMDISK (1-2/85, S. 44)
SCHIRMDISK.LISA.SOURCE
SCHIRMDISK

T.VIDEXT
VIDEXT.LISA.SOURCE
VIDEXT

GETPAS (1-2/85, S. 70)
T.GETPAS.ASS *
GETPAS.ASS
GETDOS.PASCAL.SOURCE
COPYDUPDIR.PASCAL.SOURCE

PRODOS.EDITOR.MACROS (1-2/85, S. 86)

Disk #3
(Heft 1-2, 1985, CP/M-Format)

STEUER.84 (1-2/85, S. 47)
PASS.BAS
MENUE.BAS
HELP.BAS *
A.BAS
B.BAS
C.BAS
D.BAS
E.BAS
F.BAS
G.BAS
H.BAS
I.BAS
J.BAS
K.BAS
L.BAS
M.BAS
N.BAS

Disk #4
(Heft 3 + 4, 1985)

TESTGENERATOR (3/85, S. 26)
SAETZE
BAHNFAHRT *
ZU *
TUN.UND.SOLLEN *
IRGEND *

MULTIPRECISION (3/85, S. 32)

T.WS.TRANSFER (3/85, S. 36)
WS.TRANSFER
T.WS.TRANSFER.2 *
WS.TRANSFER.2 *
GETCPM

PRIM.0.SC.SOURCE (3/85, S. 62)
PRIM.0.BIN
PRIM.1.SC.SOURCE
PRIM.1.BIN
PRIM.FP

ACCELERATOR.ABSTELLEN (3/85, S. 66)

T.WILDCARD.TEST * (3/85, S. 72)
WILDCARD.TEST1 *
T.WILDCARD.TEST2 *
WILDCARD.TEST2 *

XPLOT.DEMO (4/85, S. 18)
XPLOT.ROUTINE
T.XPLOT.ROUTINE

MENUE.GENERATOR (4/85, S. 22)

T.MACROS.65C02 (4/85, S. 31)

TERMINAL (4/85, S.36)
TERMINAL.B
T.TERMINAL.B

CAT.ARRAY (4/85, S. 44)
CAT.SAVER
EINTRAG.SUCHER
EINTRAG.ANALYSE
PRODOS.READER
T.PRODOS.READER.OBJ
PRODOS.READER.OBJ

MOUSESTUFF.PASCAL.SOURCE (4/85, S. 51)
MOUSE.ASS.PASCAL.SOURCE
TESTMOUSE.PASCAL.SOURCE
DRAWMOUSE.PASCAL.SOURCE

INALL.DATA (4/85, S. 70)
SCREEN80.DATA (4/85, S. 33)
SCREEN80.SAVER (4/85, S. 76)

Disk #5
(Heft 5, 1985, DOS-Format)

T.FM.BSP (5/85, S. 9)
FM.BSP

T.SLOTRAMDISK (5/85, S. 13)
SLOTRAMDISK
SLOTRAMDISK.HELLO

PLOT.2.0 (5/85, S. 20)
T.PLOT.B
PLOT.B
PLOT.PROTECTOR

T.CONVERT560 (5/85, S. 26)
CONVERT560
CONVERT560.DEMO

T.EDA (5/85, S. 33)
EDA

TRANSCEND.PASCAL.SOURCE (5/85, S. 36)

T.BLOCKTRACER (5/85, S. 51)
BLOCKTRACER
T.BLOCKTRACER1
BLOCKTRACER1

FORMAT.LC (5/85, S. 56)
FORMAT.LC.START
T.DISKDRIVER.DEMO
DISKDRIVER.DEMO

RANDOM.DEMO (5/85, S. 69)
COLUMN80.DEMO

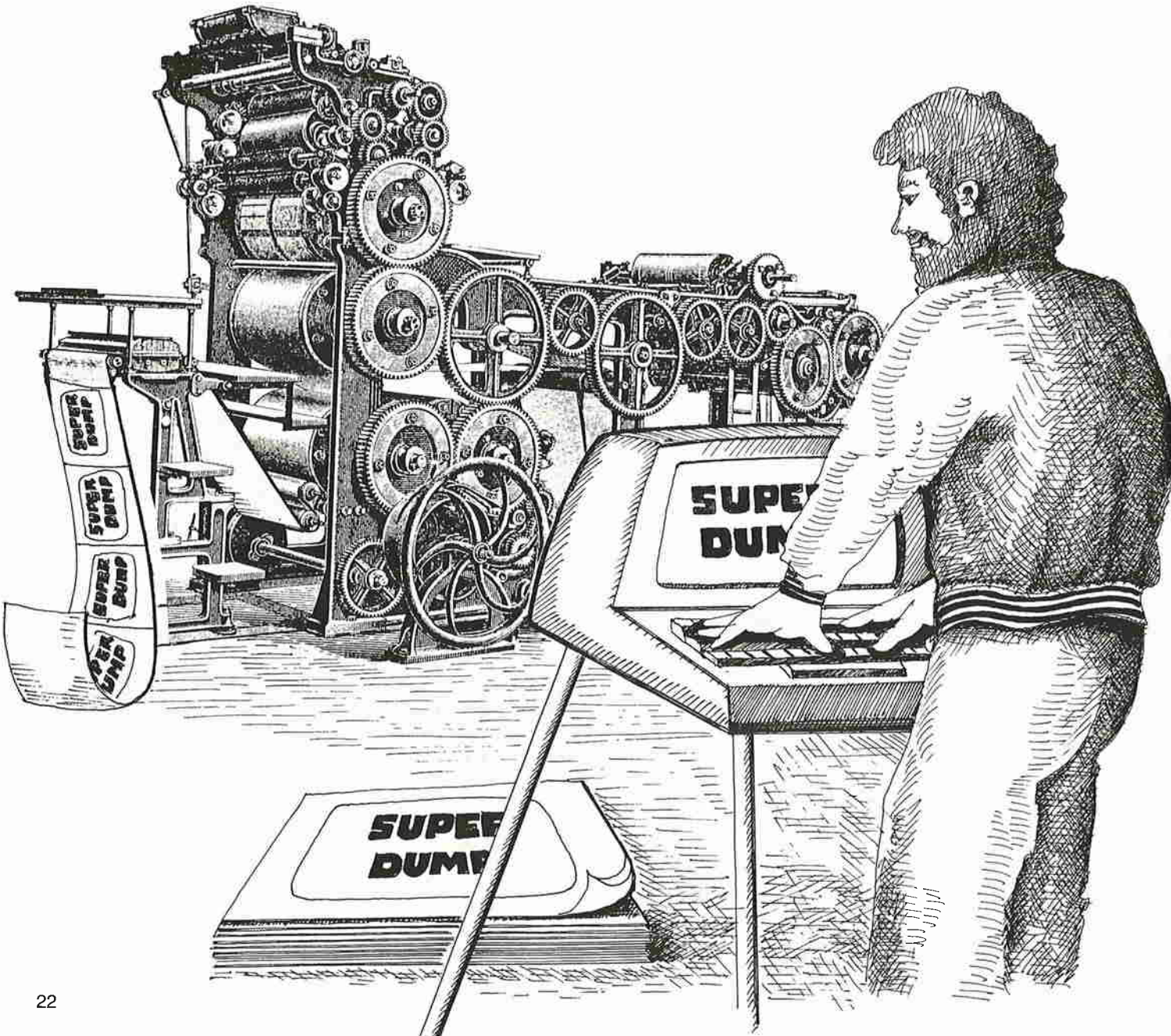
SUPERDUMP.EPSON (6/85!)
SUPERDUMP.IMAGEWRITER
SUPERDUMP.BILD
T.SUPERDUMP
SUPERDUMP
EPSON
IMAGEWRITER

Auf vielfachen Wunsch bringen wir hier eine Neuüberarbeitung des im Peeker Heft 1/84 vorgestellten Dump-Programms. Bei der Überarbeitung wurde besonderer Wert auf die Kompatibilität mit den meisten Druckern und Interfaces gelegt. Außerdem bietet SUPERDUMP eine Reihe neuer Features wie Double-Hires-Grafik, Druckern von mehreren Bildern nebeneinander usw.

Superdump

Das universelle Hires-Grafik-Dump-Programm

von Jürgen Geiß



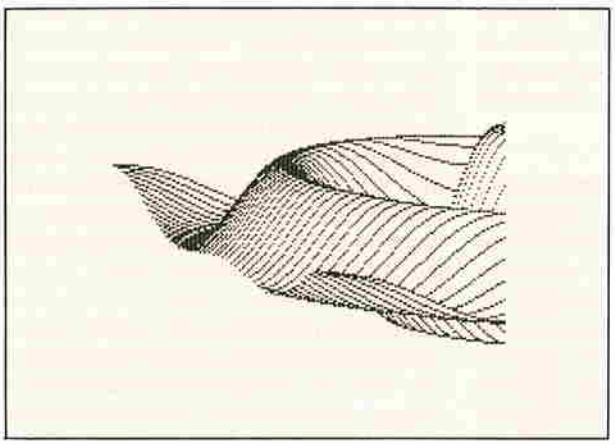


Bild 1: Linke Hälfte, nicht gestaucht

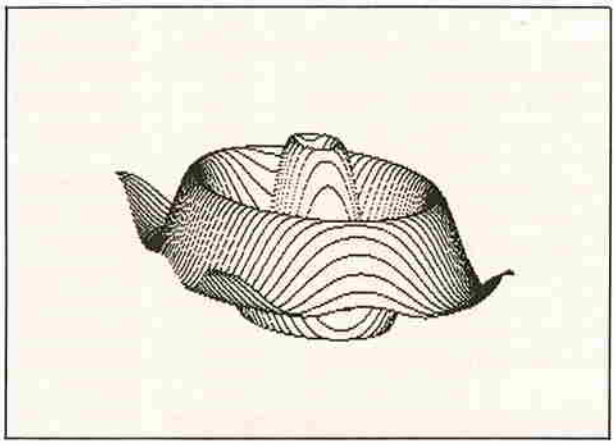


Bild 2: Beide Hälften, gestaucht

Nach Veröffentlichung der ersten Version riefen viele Leser bei der Peeker-Redaktion an und fragten den ohnehin schon gestreßten Chefredakteur Löcher in den Bauch. So traten Fragen auf wie „Bei mir wird immer nur eine Zeile gedruckt und dann nichts mehr!" oder „Ich habe keinen Epson-Drucker, was muß ich modifizieren?" usw., die Herr Stiehl natürlich nicht alle beantworten konnte. Da aber die Peeker-Leser, die nicht gerade einen Epson-Drucker mit Parallel-Interface besitzen, auch in den Genuß einer Hardcopy der Hires-Grafik kommen sollten, mußte eine ganz neue Lösung gefunden werden. Aus der Aufgabenstellung heraus, eine Hardcopy hardwareunabhängig (Drucker, Interface) ausgeben zu können, ergaben sich zwei Probleme, die gelöst werden mußten.

1. Problem: Die Druckersteuerzeichen

Wer schon einmal einen Text, der mit Steuerzeichen für Fettschrift, Kursivschrift etc. gespickt ist, auf zwei Druckern ganz verschiedenen Typs ausgegeben hat, der wird wissen, daß mindestens einer der Ausdrucke total „daneben war". Nur im seltensten Fall findet man zwei Drucker, die zueinander vollständig kompatibel sind.
Leider gibt es für Druckersteuerzeichen keine Norm außer den wenigen Zeichen wie CR (Wagenrücklauf) oder LF (Zeilenvorschub). Aber sogar hier gibt es Unterschiede. Der eine Drucker benötigt ein LF nach CR, der andere wiederum nicht.
So verhält es sich auch mit dem Umschalten auf Grafikbetrieb. Epson verlangt beispielsweise <ESC> „K" und anschließend die Anzahl der Grafik-Bytes im Low/high-Format, Apples Imagewriter dagegen <ESC> „G0280" für 280 Grafik-Bytes.

Man kann also von einem Programmierer nicht verlangen, daß er die Steuercodes aller Drucker kennt. Wenn nicht er, wer aber dann? Nehmen wir doch der Einfachheit halber einmal den Druckerbesitzer (oder auch nur den Anwender). Beim Auspacken eines Druckers findet man üblicherweise ein mehr oder weniger gutes Handbuch mit einer Tabelle aller für diesen Drucker gültigen Steuerzeichen.

Hier muß nun der Programmierer ansetzen. Er verlangt vom Benutzer, daß er im Handbuch die nötigen Steuercodes nachschlägt und die Daten dem Programm zukommen läßt. Leider gibt es ein noch schwierigeres Problem, was tage- (und nächte-)langes Brüten verursachte.

2. Problem: Das Interface

Es gibt weitaus mehr Schnittstellen denn Drucker. Es gibt deren so viele, daß der Programmierer unmöglich wissen kann, wie diese softwaremäßig anzusteuern sind. Nun besteht natürlich die Möglichkeit, von Applesoft aus mittels des PR#-Befehls die Ausgabe vom Bildschirm auf den Drucker umzulenken, um anschließend von Assembler aus über $FDED (= COUT = Character Output) die Zeichen an den Drucker zu senden. Die Sache hat allerdings einen Haken. Beim Drucken von Grafiken können alle möglichen Bitmuster (von $00 bis $FF) auftreten. Da es aber für jedes Interface wiederum Steuerzeichen gibt, um es zu programmieren, kann es vorkommen, daß der „Command-Character" auch bei einem auszudruckenden Bitmuster auftritt. Bei einem parallelen oder seriellen Interface ist dies zum Beispiel ein <Ctrl-I> CHR$(9). Dieses Grafik-Byte und einige darauffolgende würden dann vom Interface verschluckt werden.

Leider kann das Interface-Problem nicht auf den Anwender abgewälzt werden, da dieser genaue Kenntnisse über die Firmware seines Interfaces haben müßte, um zu erkennen, welche I/O-Adressen für seine Karte zuständig sind. Das aber kann man nicht von ihm verlangen. Also was tun?
Wer sich schon näher mit dem Pascal-Betriebssystem, insbesondere mit dem BIOS (Basic Input Output System) befaßt hat, findet eine Lösung dieses Problems. Pascal erkennt fast alle Schnittstellen und arbeitet problemlos mit ihnen.
Dies funktioniert wie folgt:

1. Hole das 5. und 7. Byte aus dem ROM des jeweiligen Slots (bei SUPERDUMP nur $C105 und $C107) und vergleiche die Werte mit einer Tabelle für die verschiedenen Interface-Typen. Es gibt deren 5: Parallel-, Seriell-, Communications-, Disk- und Firmware-Karten.

2. Sende das Zeichen entweder über die genormten I/O-Adressen gemäß den obigen Typen (bei Parallel- oder Communications-Karte) oder springe die genormten Adressen im Slot- bzw. im ROM-Bereich an (bei Seriell- oder Firmware-Karten).

Bei letzteren gibt es allerdings noch eine Sonderregel, die unbedingt beachtet werden muß. Da bei diesen Karten direkt in das ROM der Karte gesprungen wird, übernehmen diese wieder die weitere Steuerung und eventuelle Filterung durch den „Command-Character". Bei diesen Karten gibt es aber meistens einen Steuercode, der einmalig gesendet werden muß, um zu verhindern, daß irgendwelche Bitmuster als Steuercode interpretiert werden. Bei Apples Super-Serial-Interface (SSC) ist dies z.B. <Ctrl-I> „Z" <RE-

TURN> (siehe Handbuch), was von Applesoft aus leicht durch folgende Zeilen realisierbar wäre:
10 PRINT CHR$(4); "PR#1"
20 PRINT CHR$(9); "Z"
30 PRINT CHR$(4); "PR#0"
Wenn also nach anfänglich erfolgreichem Drucken auf einmal Buchstaben statt Grafik gedruckt werden, dann erst im Handbuch zum Interface nachschlagen, ob es nicht solch einen Code gibt und diesen dann vor dem Drucken senden.
Nun aber genug der Theorie.

Bild 3: Normales Double-Hires-Bild

Was bietet SUPERDUMP?

Da nicht alle Peeker-Leser auf das Heft 1/84 zurückgreifen können, folgt hier eine Aufzählung aller Features von **SUPERDUMP**:

- durch Angabe eines Kontrollblocks für Druckersteuerzeichen druckerunabhängig,
- 1-, 2-, oder 4facher Druck (für besonders schwarze Dumps),
- positiver oder negativer Druck (schwarz auf weiß oder weiß auf schwarz),
- horizontaler oder vertikaler Druck,
- Umrandung von Bildern,
- durch Angabe der Bildadresse (z.B. $2000 für Seite 1) speicherunabhängig,
- mehrere Bilder können nebeneinander wiederholt werden,
- Bildauschnitte können gewählt werden,
- Vergrößerung in X- und Y-Richtung (Zoom),
- Dump der Apple IIe/IIc Double-Hires-Grafik direkt aus dem Bildspeicher,
- automatischer Zeilenvorschub bei CR ein/ausschaltbar,
- Drucken eines großen Bildes durch Aufspaltung in Teilbilder,
- Richtung der Druckernadeln wählbar,
- Konvertierung von Double-Hires-Bildern nach normal und umgekehrt.

Diese Features werden durch die sog. **Übergabeparameter** in der Seite 3 ($0300) gewählt, die in **Tabelle 1** zusammengestellt sind. Es folgt eine nähere Erläuterung einzelner Parameter:

Kontrollblock-Nummer ($0300) – Wie oben erwähnt, gelten für verschiedene Drucker verschiedene Steuerzeichen. Diese müssen in einem oder mehreren Kontrollblocks (KB) abgelegt werden. Mit der Übergabe der KB-Nummer kann einer der Blöcke selektiert werden (z.B. zur Einstellung von Horizontal- oder Vertikaldruck). Der Aufbau eines Kontrollblocks sieht wie folgt aus:

Die Länge eines KBs beträgt 64 Bytes. Diese leitet sich ab aus der Länge von 4 Strings zu je 15 Zeichen, an deren Beginn das Länge-Byte stehen muß. Die 4 Strings haben folgende Bedeutung:
– String #1 = Pinit: Beinhaltet die Initialisierungssteuerzeichen wie Zeilenabstand, unidirektionaler Druck etc., die zu Beginn eines Druckvorgangs einmalig gesendet werden müssen. Achtung: Der Zeilenabstand muß bei horizontalem Druck 8 Nadeln, bei vertikalem Druck 7 Nadeln betragen.
– String #2 = Pline: Steuerzeichen zum Umschalten des Druckers auf den Grafikbetrieb. Diese Zeichen werden vor jeder Druckzeile ausgegeben.
– String #3 = Plf1: Steuerzeichen für 1/216" oder 1/144" Zeilenvorschub (wird nur bei 2- und 4fach-Druck benutzt).
– String #4 = Pexit: abschließende Steuerzeichen wie bidirektionaler Druck, Zeilenabstand auf alten Wert etc.

In **Tabellen 2** sind die Werte für die Epson-Drucker und den Imagewriter wiedergegeben. Diese Hex-Dumps sollten durch BSAVE EPSON (oder IMAGEWRITER), A$9000, L$80 gespeichert werden.
Bei der Erstellung eigener Kontrollblöcke ist zu beachten, daß sich die Parameterblöcke nicht mit dem Assembler- ($8000-$86C6) oder einem Applesoft-Programm (ab $0800) überlappen.

Mehrfachdruck ($0304) – Bei 2- und 4fachem Druck wird versucht, das Papier um 1/216" oder 1/144" vorwärts zu schieben, um nochmals die gleiche Zeile zu drucken. Der aufmerksame Leser, der einen Epson-Drucker besitzt, wird vielleicht bemerkt haben, daß bei Pinit der Zeilenabstand des Druckers nicht wie erwartet auf 24/3 = 8 Bits initialisiert wurde, sondern nur auf 23/3 Nadeln. Die restliche 1/3 Nadel besorgt dann der Plf1-String. Auch bei einfachem Druck wird dieser String zum Drucker gesendet, wenn er nicht die Länge 0 hat. Es wird dann lediglich der fehlende Zeilenvorschub nachgeholt. Wer will, kann auch in **Tabelle 2** den Abstand auf 24:3 = 8 Bits stellen und den Plf1-String auf 0 setzen. Für vertikalen Druck gilt dasselbe.

Teilbilder ($0322) – An dieser Stelle möchte ich mich ganz herzlich für die positive Kritik an der Erstversion des Dump-Programms von Herrn N.G. Barbieri bedanken, bei dessen Graf-quattro-Editor (Peeker Heft 4/85) diese Neuversion durch den DiffPics-Parameter (wie er im Assemblerlisting genannt wird) bestimmt Verwendung finden wird. Beim Drucken seiner 560 * 384-Punkte-Grafiken muß dann wie folgt vorgegangen werden: Laden der beiden Teilbilder 1 und 2 in Hires-Seite 1 und 2, Parameter wie im Beispiel angegeben setzen, dann die Bilder 3 und 4 laden und nochmals drucken (wie oben).

Kommando-Byte ($0326) – Anwendung für Kommando 1 und 2: Vor dem Abspeichern eines Double-Hires-Bildes muß dieses aus der 64K-Karte in das Main-RAM geschoben werden (1); nach dem Laden entsprechend umgekehrt (2).

Modifizierung und Anwendung der Programme

Das Assembler-Programm muß mit „BRUN SUPERDUMP" gestartet werden, nachdem es mit „BSAVE SUPERDUMP, A$8000, L$6C7" gespeichert wurde, und kann dann von Applesoft aus über den &-Vektor aufgerufen werden.
Wegen der Länge des Quelltextes, der mehr als 1000 Zeilen umfaßt, kann hier nur ein Hex-Dump abgedruckt werden.
Für all diejenigen, die die Paramaterübergabe in Seite 3 nicht selbst übernehmen wollen, gibt es die Applesoft-Rahmenprogramme **SUPERDUMP.IMAGEWRITER** und **SUPERDUMP.EPSON**, die dies besorgen. Für andere Drucker müssen folgende Zeilen geändert werden:

1000: PF$ ist der Printerfile, in dem die Grafiksteuerzeichen stehen. Dieser File muß vorher, am besten mit dem Druckernamen, abgespeichert werden. Dabei spielt die Adresse keine Rolle, denn diese wird in Zeile 1500-1510 nach dem Laden automatisch ermittelt. *Achtung:* Dies gilt nur für DOS 3.3 und nicht für ProDOS! Am

besten benutzt man Block #0 des Printerfiles für horizontalen und Block #1 für vertikalen Druck (siehe Tabelle 2).

1010: DG ist der Grundeinstellungswert des Grafikmodus des benutzten Druckers, wenn ein Double-Hires-Bild mittels Grundeinstellung gedruckt werden soll. Da ein Double-Hires-Bild in X-Richtung doppelt so eng auf dem Bildschirm erscheint, sollte auch für den Drucker ein entsprechend enger Grafikmodus gewählt werden. Beim Epson sollte dieser Wert 2 (960-Punkte-Grafik), beim Imagewriter 4 (1152-Punkte-Grafik) sein.

1020: NG ist der Grundeinstellungswert für die Richtung der Drucknadeln. Für einen Epson-Drucker muß dieser Wert 0 sein, für Imagewriter 1.

3280–3350: Diese Zeilen haben folgende Bedeutung: Sie stellen dem Benutzer die Anzahl der Grafik-Bytes pro Druckzeile in 3 verschiedenen Formaten zur Verfügung:

1. BT ist die Anzahl der Bytes als Integer-Zahl
2. BL, BH ist die Anzahl der Bytes im Low/high-Format (für Epson-Drucker etc.)
3. BT$ ist die Anzahl der Bytes im Stringformat (für Imagewriter etc.)

3400 - Ende: Menü der verschiedenen Grafikmodi für einen speziellen Drucker (siehe entsprechende Werte für Epson und Imagewriter). Die Modi entnimmt man den Druckerhandbüchern. Anschließend muß die Anzahl der Grafik-Bytes in den entsprechenden Parameterblock gepokt werden. Dies besorgen die letzen 2 Zeilen.

Achtung: Die Länge des Applesoft-Programms inklusive Variablen sollte $2000 nicht überschreiten, wenn aus Seite 1 gedruckt werden soll.

Hinweis für Assembler-Programmierer, die das Applesoft-Rahmenprogramm nicht benutzen wollen: Die Parameter auf Seite 3 müssen selbst an das Druckprogramm übergeben werden. Diese können aber vorher schon abgespeichert worden sein und brauchen dann nur noch – wie auch der Printerfile für die Druckersteuerzeichen – geladen werden.

Für die Benutzung der Applesoft-Rahmenprogramme sind in **Tabellen 3** die Eingaben für die einzelnen Beispielbilder wiedergegeben.

Und nun viel Spaß beim Drucken. Übrigens: Wenn nicht alles gleich auf Anhieb klappt, nicht verzagen. Erst probieren, dann anrufen.

Hinweis: Wegen der großen Nachfrage wurde SUPERDUMP in den Versionen für Imagewriter und Epson bereits in die Peeker-Sammeldisk #5 aufgenommen

SUPERDUMP

```
BSAVE SUPERDUMP, A$8000, L$6C7

$8000: A9 11 8D F6 03 A9 80 8D
$8008: F7 03 20 C4 85 20 FB 85
$8010: 60 AD 26 03 F0 1D C9 01
$8018: D0 03 4C E3 84 C9 02 D0
$8020: 03 4C E9 84 C9 03 D0 03
$8028: 4C 48 85 C9 04 D0 03 4C
$8030: 85 85 60 AD 1E 03 F0 12
$8038: 20 6D 83 AD 54 C0 AD 57
$8040: C0 AD 5E C0 8D 01 C0 8D
$8048: 0D C0 AD 0A 03 F0 03 20
$8050: F6 83 20 AA 83 AD 08 03
$8058: F0 03 4C DD 81 20 E1 83
$8060: AC 16 03 8C 31 03 A9 01
$8068: 8D 39 03 20 8F 80 AD 39
$8070: 03 CD 1C 03 F0 06 EE 39
$8078: 03 4C 6B 80 AD 31 03 18
$8080: 69 08 8D 31 03 CD 18 03
$8088: 90 DC F0 DA 4C 61 83 20
$8090: 2F 83 20 C3 80 20 3E 83
$8098: CE 3A 03 D0 F5 20 9C 81
$80A0: 20 5D 83 AE 04 03 E0 01
$80A8: F0 07 E0 02 D0 11 20 9C
$80B0: 81 20 2F 83 20 C3 80 20
$80B8: 3E 83 CE 3A 03 D0 F5 20
$80C0: 57 83 60 A9 00 8D 3D 03
$80C8: AD 12 03 8D 38 03 AC 13
$80D0: 03 8C 34 03 AD 31 03 8D
$80D8: 37 03 A9 08 8D 36 03 AD
$80E0: 37 03 20 7D 84 20 C9 84
$80E8: AE 38 03 4A CA 10 FC 2E
$80F0: 32 03 CE 36 03 F0 1F AD
$80F8: 37 03 CD 18 03 F0 06 EE
$8100: 37 03 4C DF 80 A9 FF 0E
$8108: 32 03 0A CE 36 03 D0 F7
$8110: 8D 3F 03 20 B2 81 A0 08
$8118: AD 32 03 4A 08 AE 1C 03
$8120: 8E 3C 03 AE 1C 03 8E 36
$8128: 03 A2 00 3E 00 02 E8 CE
$8130: 36 03 D0 F7 28 08 CE 3C
$8138: 03 D0 E8 28 88 D0 DC AE
$8140: 39 03 8E 37 03 A0 08 18
$8148: AE 1C 03 CA 7E 00 02 CA
$8150: 10 FA 2A 88 D0 F1 CE 37
$8158: 03 D0 EA 4D 06 03 20 06
$8160: 83 AE 1A 03 20 18 83 CA
$8168: D0 FA AC 34 03 CC 15 03
$8170: D0 08 AD 38 03 CD 14 03
$8178: F0 13 EE 38 03 AD 38 03
$8180: C9 07 D0 06 C8 A9 00 8D
$8188: 38 03 4C D1 80 EE 3D 03
$8190: AD 3D 03 CD 22 03 F0 03
$8198: 4C C8 80 60 AD 04 03 C9
$81A0: 02 D0 0E 20 2F 83 20 C3
$81A8: 80 20 3E 83 CE 3A 03 D0
$81B0: F5 60 AD 06 03 F0 0C AD
$81B8: 3F 03 4D 06 03 0D 32 03
$81C0: 8D 32 03 AD 32 03 60 AD
$81C8: 04 03 C9 02 D0 0E 20 2F
$81D0: 83 20 39 82 20 3E 83 CE
$81D8: 3A 03 D0 F5 60 20 E1 83
$81E0: AC 15 03 8C 34 03 A9 01
$81E8: 8D 39 03 20 2F 83 20 39
$81F0: 82 20 3E 83 CE 3A 03 D0
$81F8: F5 20 C7 81 20 5D 83 AE
$8200: 04 03 E0 01 F0 07 E0 02
$8208: D0 11 20 C7 81 20 F0 2F 83
$8210: 20 39 82 20 3E 83 CE 3A
$8218: 03 D0 F5 20 57 83 AD 39
$8220: 03 CD 1A 03 F0 06 EE 39
$8228: 03 4C EB 81 AD 34 03 A8
$8230: 88 CD 13 03 D0 AD 4C 61
$8238: 83 A9 00 8D 3D 03 AD 16
$8240: 03 8D 31 03 20 7D 84 AC
$8248: 34 03 20 C9 84 8D 32 03
$8250: CC 15 03 D0 1C A9 06 38
$8258: ED 14 03 F0 14 AA E8 A9
$8260: FF 4A CA D0 FC 8D 3F 03
$8268: 2D 32 03 8D 32 03 20 B2
$8270: 81 AD 32 03 AC 13 03 CC
$8278: 34 03 D0 17 AE 12 03 F0
$8280: 12 A9 FF 0A CA D0 FC 8D
$8288: 3F 03 2D 32 03 8D 32 03
$8290: 20 B2 81 A0 07 4A 08 AE
$8298: 1A 03 8E 3B 03 AE 1A 03
$82A0: 8E 36 03 A2 00 3E 00 02
$82A8: E8 CE 36 03 D0 F7 28 08
$82B0: CE 3B 03 D0 E8 28 88 D0
$82B8: DC AE 39 03 8E 37 03 A0
$82C0: 07 18 AE 1A 03 CA 7E 00
$82C8: 02 CA 10 FA 2A 88 D0 F1
$82D0: CE 37 03 D0 EA 4D 06 03
$82D8: 29 7F 20 06 83 AE 1C 03
$82E0: 20 18 83 CA D0 FA AE 31
$82E8: 03 EE 31 03 AD 31 03 EC
$82F0: 18 03 B0 03 4C 44 82 EE
$82F8: 3D 03 AD 3D 03 CD 22 03
$8300: F0 03 4C 3E 82 60 AE 24
$8308: 03 F0 0C A2 08 6A 2E 3E
$8310: 03 CA D0 F9 AD 3E 03 60
$8318: 08 8D 32 03 8A 48 98 48
$8320: AD 32 03 20 46 86 68 A8
$8328: 68 AA AD 32 03 28 60 A9
$8330: 0D 20 18 83 20 E5 83 AD
$8338: 0E 03 8D 3A 03 60 AD 10
$8340: 03 AA AD 11 03 A8 D0 03
$8348: 8A F0 0B A9 00 20 18 83
$8350: CA D0 FA 88 10 F7 60 A9
$8358: 0A 20 18 83 60 20 E9 83
$8360: 60 20 ED 83 AD 1E 03 F0
$8368: 03 20 86 83 60 0E 18 C0
$8370: 2E 40 03 0E 1C C0 2E 40
$8378: 03 0E 1D C0 2E 40 03 0E
$8380: 1F C0 2E 40 03 60 4E 40
$8388: 03 B0 03 8D 0C C0 4E 40
$8390: 03 B0 03 8D 56 C0 4E 40
$8398: 03 8D 54 C0 90 03 8D 55
$83A0: C0 4E 40 03 B0 03 8D 00
$83A8: C0 60 AD 00 03 85 04 A9
$83B0: 00 85 05 A2 06 06 04 26
$83B8: 05 CA D0 F9 18 AD 02 03
$83C0: 65 04 85 04 AD 03 03 65
$83C8: 05 85 05 60 B1 04 F0 10
$83D0: 8E 33 03 AA C8 B1 04 20
$83D8: 18 83 CA D0 F7 AE 33 03
$83E0: 60 A0 00 F0 E7 A0 10 D0
$83E8: E3 A0 20 D0 DF A9 0D 20
$83F0: 18 83 A0 30 D0 D6 A9 00
$83F8: 20 7D 84 A0 27 AD 1E 03
$8400: F0 02 A0 4F A9 FF 4D 06
$8408: 03 20 D5 84 88 10 FA A9
$8410: BF 20 7D 84 A0 27 AD 1E
$8418: 03 F0 02 A0 4F A9 FF 4D
$8420: 06 03 20 D5 84 88 10 FA
$8428: A2 01 8A 20 7D 84 A0 00
$8430: 20 C9 84 8D 32 03 A9 01
$8438: 4D 06 03 30 05 0D 32 03
$8440: D0 03 2D 32 03 8D 32 03
$8448: AD 32 03 20 D5 84 A0 27
$8450: AD 1E 03 F0 02 A0 4F 20
$8458: C9 84 8D 32 03 A9 40 4D
$8460: 06 03 C9 40 D0 05 0D 32
$8468: 03 D0 03 2D 32 03 8D 32
$8470: 03 AD 32 03 20 D5 84 E8
$8478: E0 BF 90 AE 60 48 29 C0
$8480: 85 02 4A 4A 05 02 85 02
$8488: 68 85 03 0A 0A 0A 26 03
$8490: 0A 26 03 0A 66 02 18 A5
$8498: 02 6D 0C 03 85 02 A5 03
$84A0: 29 1F 6D 0D 03 85 03 AE
$84A8: 3D 03 F0 0A 18 A5 03 69
$84B0: 20 85 03 CA D0 F6 60 AD
$84B8: 1E 03 F0 0C 98 4A A8 90
$84C0: 04 8D 54 C0 60 8D 55 C0
$84C8: 60 8C 35 03 20 B7 84 B1
$84D0: 02 AC 35 03 60 8C 35 03
$84D8: 48 20 B7 84 68 91 02 AC
$84E0: 35 03 60 A2 20 A0 40 D0
$84E8: 04 A2 40 A0 20 86 01 84
$84F0: 03 A2 20 A0 00 84 00 84
$84F8: 02 20 6D 83 AD 57 C0 8D
$8500: 01 C0 AD 55 C0 B1 00 91
$8508: 02 C8 D0 F9 E6 01 E6 03
```

```
$8510: CA D0 F2 20 86 83 60 A9
$8518: 00 8D 3D 03 8D 0C 03 A9
$8520: 20 8D 0D 03 A9 01 8D 1E
$8528: 03 20 6D 83 AD 57 C0 8D
$8530: 01 C0 60 8E 33 03 20 7D
$8538: 84 AE 33 03 18 A5 02 85
$8540: 04 A5 03 69 20 85 05 60
$8548: 20 17 85 A2 00 8A A0 27
$8550: 8C 34 03 A0 4F 8C 35 03
$8558: 20 33 85 AC 35 03 20 C9
$8560: 84 AC 34 03 91 04 CE 35
$8568: 03 88 8C 34 03 10 EC A0
$8570: 27 20 C9 84 8D 54 C0 91
$8578: 02 88 10 F5 E8 E0 C0 D0
$8580: CC 20 86 83 60 20 17 85
$8588: A2 00 8A 20 33 85 A0 00
$8590: 8D 54 C0 B1 02 20 D5 84
$8598: C8 C0 28 D0 F3 8C 35 03
$85A0: A0 00 8C 34 03 AC 34 03
$85A8: B1 04 AC 35 03 20 D5 84
$85B0: EE 34 03 C8 8C 35 03 C0
$85B8: 50 D0 EA E8 E0 C0 D0 CA
$85C0: 20 86 83 60 A9 00 8D 30
$85C8: 03 A2 05 AD 05 C1 DD BD
$85D0: 86 D0 08 AD 07 C1 DD C1
$85D8: 86 F0 06 CA E0 02 B0 EB
$85E0: 60 E0 04 D0 12 AD 0B C1
$85E8: C9 01 D0 0B A2 06 AD 0C
$85F0: C1 C9 31 D0 02 A2 01 8E
$85F8: 30 03 60 AD 30 03 C9 01
$8600: F0 24 C9 03 F0 0D C9 04
$8608: F0 14 C9 05 F0 17 C9 06
$8610: F0 14 60 A9 03 8D 9E C0
$8618: A9 15 8D 9E C0 60 20 36
$8620: 86 20 00 C8 60 60 48 20
$8628: 36 86 AD 0D C1 85 00 AC
$8630: F8 06 68 6C 00 00 A2 C1
$8638: A0 10 86 01 8C F8 06 AD
$8640: FF CF AD 00 C1 60 AE 20
$8648: 03 F0 04 C9 0A F0 17 AE
$8650: 30 03 E0 01 F0 53 E0 03
$8658: F0 0D E0 04 F0 15 E0 05
$8660: F0 1D E0 06 F0 22 60 AA
$8668: AD 9E C0 29 02 F0 F9 8E
$8670: 9F C0 60 48 20 36 86 68
$8678: 9D B8 05 20 AA C9 60 AE
$8680: C1 C1 30 FB 8D 90 C0 60
$8688: 48 A9 00 20 A1 86 20 36
$8690: 86 AD 0F C1 4C 2D 86 48
$8698: 20 36 86 AD 10 C1 4C 2D
$86A0: 86 48 20 97 86 68 90 F9
$86A8: 60 AA A0 10 B9 89 C0 29
$86B0: 10 F0 F9 B9 89 C0 29 40
$86B8: D0 F2 8A 99 88 C0 60 03
$86C0: 18 38 48 3C 38 18 48 00
```

SUPERDUMP.IMAGEWRITER

```
1000 PF$ = "IMAGEWRITER": REM Parameter-File
1010 DG = 4: REM Grundeinstellung für Double-Hires
1020 NG = 1: REM Grundeinstellung für Nadel
1030 ST = 768: REM Start der ParameterÜbergabe
1040 KB = ST + 00: REM Nummer des Kontrollblockes
1050 KA = ST + 02: REM Adresse des Kontrollblockes
1060 IM = ST + 04: REM 1-, 2- oder 4-fach Druck
1070 XO = ST + 06: REM exclusives Oder für jedes Byte
1080 DI = ST + 08: REM Richtung
1090 FR = ST + 10: REM Rand
1100 AD = ST + 12: REM Grafikadresse ($2000 für Seite 1)
1110 RF = ST + 14: REM Wiederholung fÜr gleiche Bilder
1120 RD = ST + 16: REM Distanz zwischen gleichen Bildern
1130 LE = ST + 18: REM linker Bildrand (0..559)
1140 RI = ST + 20: REM rechter Bildrand (0..559)
1150 TP = ST + 22: REM oberer Bildrand (0..191)
1160 BO = ST + 24: REM unterer Bildrand (0..191)
1170 XF = ST + 26: REM Dehnung in X-Richtung (1..31)
1180 YF = ST + 28: REM Dehnung in Y-Richtung (1..31)
1190 HI = ST + 30: REM Double-Hires-Grafik fÜr Apple //e
1200 AL = ST + 32: REM automatisches LF nach CR
1210 DP = ST + 34: REM Anzahl verschiedener Bilder
1220 NE = ST + 36: REM Richtung der Drucknadeln
1230 CO = ST + 38: REM Kommando fÜr Assemblerroutine
1240 IN = ST + 48: REM Interface-Typ
1245 D$ = CHR$(4)
1250 REM
1260 TEXT : PRINT CHR$ (12): HOME
1270 PRINT "************************"
1280 PRINT "*  S U P E R D U M P  *"
1290 PRINT "*   von Jürgen Geiß   *"
1300 PRINT "*      März 1985      *"
1310 PRINT "************************"
1320 PRINT : PRINT D$"BRUN SUPERDUMP"
1330 PRINT : PRINT "Augenblicklicher Drucker: ";PF$;
1340 PRINT : PRINT "Schnittstelle: ";
1350 ON PEEK (IN) GOTO 1360,1370,1380,1390,1400,1410
1360 S$ = "Super-Serielle": GOTO 1420
1370 S$ = "Disk-Controller": GOTO 1420
1380 S$ = "Communications": GOTO 1420
1390 S$ = "Serielle": GOTO 1420
1400 S$ = "Parallele": GOTO 1420
1410 S$ = "Firmware"
1420 PRINT S$;" Karte";
1430 PRINT : PRINT "Apple //e Double-Hires-Grafik?
     (J/N)";: GOSUB 2150
1440 IF A$ < > "J" GOTO 1470
1450 POKE 49153,0: POKE 49165,0:X = PEEK (49246): REM
     Softswitches fÜr Apple //e
1460 PRINT : PRINT D$"PR#3";:D = 1
1470 PRINT : PRINT "Bild speichern? (J/N)";: GOSUB 2150
1480 IF A$ < > "J" GOTO 1520
1490 L$ = "L$2000"
1500 IF D = 1 THEN POKE C0,1: & :L$ = "L$4000"
1510 PRINT : PRINT D$"BSAVE BILD,A$2000,";L$
1520 PRINT : PRINT D$"BLOAD";PF$
1530 P1 = PEEK (43634): REM Adresse des zuletzt
1540 P2 = PEEK (43635): REM geladenen Parameterblockes
1550 PB = P1 + 256 * P2
1560 P = 1
1570 HOME : VTAB (22)
1580 PRINT "Grafik schon im Speicher? (J/N)";: GOSUB 2150
1590 IF A$ = "N" GOTO 1700
1600 IF A$ = "J" GOTO 1800
1610 GOTO 1570
1620 POKE - 16300,0: POKE - 16301,0: POKE - 16297,0: POKE
     - 16304,0
1630 HOME : VTAB (22)
1640 FOR I = ST TO ST + 39: POKE I,0: NEXT : REM Parameter
     lÖschen
1650 GOSUB 2190: REM Grundeinstellung
1660 PRINT "Grundeinstellung? (J/N)";: GOSUB 2150
1670 IF A$ = "J" GOTO 2020
1680 TEXT
1690 GOSUB 2400: GOSUB 2480: GOSUB 2550: GOSUB 2620: GOSUB
     2690: GOSUB 2760: GOSUB 2830: GOSUB 2900: GOSUB 3380:
     GOTO 2020
1700 HOME : VTAB (22)
1710 PRINT "Diskette mit Grafikfiles einschieben": PRINT
     "und Taste drücken";
1720 GET A$
1730 TEXT : PRINT : PRINT D$"CATALOG": PRINT
1740 INPUT "Welche Grafik laden? ";A$
1750 IF A$ = "" THEN A$ = "BILD"
1760 HGR
1770 HOME : VTAB (22)
1780 PRINT D$"BLOAD";A$
1790 IF D = 1 THEN POKE C0,2: & : REM verschiebe von Seite 2
     nach Aux RAM
1800 POKE - 16300,0: POKE - 16301,0: POKE - 16297,0: POKE
     - 16304,0
1810 P = 1
1820 HOME : VTAB (22)
1830 PRINT "Ist dies das richtige Bild? (J/N)";: GOSUB 2150
1840 IF A$ = "J" GOTO 1620
1850 IF A$ = "N" GOTO 1870
1860 GOTO 1820
1870 HOME : VTAB (22)
1880 PRINT "Wollen Sie die andere Grafikseite": PRINT
     "sehen? (J/N)";: GOSUB 2150
1890 IF A$ = "J" GOTO 1920
1900 IF A$ = "N" GOTO 2000
1910 GOTO 1870
1920 POKE - 16299,0
1930 POKE - 16302,0
1940 P = 2
1950 GET A$
1960 PRINT
1970 POKE - 16300,0
1980 POKE - 16301,0
1990 GOTO 1820
2000 POKE - 16303,0
2010 GOTO 1700
2020 POKE - 16302,0
```

Für Ihre Unterlagen

Abonnement bestellt

am: _____

Vertrauensgarantie:
Ich habe davon Kenntnis genommen, daß ich die Bestellung schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag, Postfach 10 28 69, 6900 Heidelberg 1 innerhalb von 7 Tagen widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels).

peeker
Leserservice
Postfach 10 28 69
6900 Heidelberg 1

Für Ihre Unterlagen

Folgende Bücher bestellt:

_____

_____

_____

_____

_____

am: _____

bei:

peeker
Versandbuchhandlung
Postfach 10 28 69
6900 Heidelberg 1

Für Ihre Unterlagen

Folgende Disketten
und Programme bestellt:

_____

_____

_____

_____

_____

am: _____

bei:

peeker
Softwareabteilung
Postfach 10 28 69
6900 Heidelberg 1

peeker
MAGAZIN FÜR APPLE-COMPUTER

Abo-Karte

Ja, ich möchte **peeker** abonnieren.
Liefern Sie mir **peeker** ab Ausgabe ..... (1985 erscheinen 11 Ausgaben – 1 Doppelnummer) zum Jahresbezugspreis von DM 72,– (Inland) incl. MwSt. Die Lieferung erfolgt frei Haus. Porto, Verpackung und Zustellgebühren übernimmt der Verlag. Der Jahresbezugspreis für das Ausland beträgt DM 72,– incl. MwSt., zzgl. DM 16,80 Versandspesen.

Ich wünsche jährliche Berechnung durch:

☐ Verlagsrechnung ☐ Abbuchung von meinem Bank- bzw. Postscheckkonto

Bank / PschA

Bankleitzahl Kto.-Nr.

Datum Unterschrift

peeker
MAGAZIN FÜR APPLE-COMPUTER

Buch-Shop

Bitte senden Sie mir gegen Rechnung folgende Bücher:

Menge	Autor, Titel	à DM	gesamt DM

Datum Unterschrift

peeker
MAGAZIN FÜR APPLE-COMPUTER

Software-Karte

Bitte senden Sie mir
gegen Rechnung folgende Apple-Programme:

☐ Peeker-Sammeldiskette, einzeln
Disk# _____, Disk# _____
Disk# _____, Disk# _____
Preis je Disk DM 28,– (einzeln)

☐ Peeker Sammeldiskette,
im Fortsetzungsbezug
ab Disk # _____
(Mindestbezug 6 Disketten)
Preis je Disk DM 20,–

☐ Apple DOS 3.3, Begleitdiskette, DM 28,–
☐ Apple ProDOS, Band 1, Begleitdiskette, DM 28,–
☐ Apple ProDOS, Band 2, Begleitdiskette, DM 28,–
☐ Apple Assembler, Begleitdiskette, DM 28,–
☐ ProDOS-Editor 1.0, Programm, DM 98,–
☐ MMU 2.0, Programm, DM 98,–
☐ INPUT 2.0, Programm, DM 98,–
☐ Softbreaker 1.0, Programm, DM 48,–
☐ DB-Meister, Programm, DM 290,–
☐ Superplot, Programm, DM 48,–
☐ Superquick, Programm, DM 48,–

Datum Unterschrift

Abo-Karte

Name

Firma

Abteilung

Straße

PLZ/Ort

Vertrauensgarantie:
Ich habe davon Kenntnis genommen, daß ich die Bestellung schriftlich durch Mitteilung an den Dr. Alfred Hüthig Verlag, Postfach 10 28 69, 6900 Heidelberg 1 innerhalb von 7 Tagen widerrufen kann. Zur Fristwahrung genügt die rechtzeitige Absendung des Widerrufs (Datum des Poststempels).

Datum

Unterschrift

Verlagshinweis:
Das Abonnement verlängert sich zu den jeweils gültigen Bedingungen um ein Jahr, wenn es nicht 2 Monate vor Jahresende schriftlich gekündigt wird.

POSTKARTE

peeker
Leserservice

Postfach 10 28 69

6900 Heidelberg 1

Buch-Shop

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

peeker
Versandbuchhandlung

Postfach 10 28 69

6900 Heidelberg 1

Software-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

POSTKARTE

peeker
Softwareabteilung

Postfach 10 28 69

6900 Heidelberg 1

INPUT 2.0

Ein Bildschirm-Maskengenerator für DOS 3.3 und ProDOS

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1021-5

„Input 2.0" liegt wahlweise in der Bank 1 oder Bank 2 der Language Card und wird durch einen kurzen Driver in den unteren 48K aufgerufen.
Für jedes Feld der Bildschirmmaske lassen sich u. a. definieren: Feldlänge (bis zu 255 Zeichen) – Vtab – Htab – Datentyp (insgesamt 8 Typen) – Scrollflag (starre oder dynamische Maske) – Ctrlflag – Füllflag – Löschflag – Bildschirmflag (40- oder 80-Z-Darstellung). Innerhalb eines Eingabefeldes besteht jeder denkbare Redigierkomfort (Insert, Delete, Rubout, Restore usw.).
Gerätevoraussetzung: Apple IIe oder IIc; ferner Apple II+ im 40-Zeichenmodus

MMU 2.0

Memory Managements Utilities

für die Apple IIe 64K-Karte DOS 3.3 (und ProDOS)

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7787-1023-1

Insgesamt enthält die neue „MMU 2.0"-Diskette über 25 Programme, die neue Einsatzmöglichkeiten für die Extended 80 Column Card (erweiterte 80-Z-Karte = 64K-Karte für den Apple IIe) erschließen. Ein Teil der Programme laufen auch auf dem Apple II Plus, doch ist „MMU 2.0" primär für 64K-Karte-Besitzer gedacht.

Gerätevoraussetzung: Apple IIe mit 64K-Karte oder IIc

Softbreaker 1.0

Eine softwaremäßige Interrupt-Utility für die Apple IIe 64K-Karte

von U. Stiehl

1984, Diskette und Manual, DM 48,–
ISBN 3-7785-1022-3

Softbreaker ist ein Assemblerprogramm, mit dessen Hilfe Programme, die sich von der 64K-Karte (= Extended 80 Column Card für den Apple IIe) starten lassen, unterbrochen, gespeichert, geladen und exakt an der Stelle der Unterbrechung fortgeführt werden können. Dadurch ist es auch möglich, Sicherungskopien von sogenannten kopiergeschützten Programmen herzustellen.
Mit Softbreaker unterbrochene Programme werden komplett, d. h. die ganzen 64K einschließlich Language Card, in nur ca. 11 Sekunden auf einer formatierten Diskette gesichert.
Gerätevoraussetzung: Apple IIe mit 64K-Karte

Hüthig Software Service,
Postfach 10 28 69, D-6900 Heidelberg

peeker Börse

Verkauf Software

TEAC 55F 600,– 55B 550,– * SSDD 10 St 42,– 96 tpi DS 78,– Lochverst. europ. Markenw. * Box Rglas. Schl. für 40–42,– für 80–49,– * Monitore 18Mhz 12" entsp. gr. ab 298,–. **VICO**, Selchower Str. 31, 1000 Bln 44

APPLE-Nachbau mit 16K, Z80, VIA-Karte, 2 Disk-Laufwerke, Monitor und Intellig. Keyboard zu verk., Tel. 040/701 31 01 nach 18 Uhr.

Flache Siemens/Rafi Tastatur mit 64 Zusatz-Funktionstasten und HEX Block. F-Tasten verkettbar bis 1800 Zeichen. Standartbelegung Basic + Wordstar. VHS 920. Eproms 2716 á 8,50. Tel. 0 62 51/6 97 42.

PrinterPlotter Adcomp X132SP mit V24 Schnittstelle. Neupreis 5500 VHS 2400 DM. Eproms 2716 á 8,50 DM, Tel. 0 62 51/6 97 42.

*** **Apple II Supermodem** *** V21, V23 und Bell, 300–1200 Baud komplett auf einer Karte, mit Software startfertig nur 398 DM. Rolf Kiupel, Tel. 04 31/55 54 27.

Verk. neuen **EPROM-Burner** NP 373,– (Fa. Springmann) für nur 320,– VHB; Frank Fuhrmann, Tel. 0 72 31/2 12 73

Verkaufe **100 % Apple Komp.** evtl. mit Zubehör, Preis VB Tel.: 0 61 09/6 48 99.

Fernschreiberinterface am Gameport m. Programm DM 79,– P. Benner, Hubertusstr. 131, 4150 Krefeld

Z80B-Karte 6MHZ DM 600,–, 8 Zoll Controler DM 500,–, 8 Laufw. DS/DD Sliml. 1,2MB DM 700,–, Tel. 02 21/60 50 88 ab 18.00 Uhr.

Verkauf Hardware

****
Landw. Spitzenprogramme
Ab 90 DM Infomappe (3 DM)
Fuer APPLE II
A. Wachendorf,
2814 Engeln 30
****

160-Track-Kopierprogramm für normale 16-Sekt.-Disks mit komf. Fehlerbehandlung Maschinenspra. B. Rüter, Rahden. Str. 65, 4955 Hille

– – – STOCKMASTER II – – – Das Apple-Programm für echte **Börsengewinne.** TÖNGI, COMPUTER-PRAXIS Aspeltstr. 4, D-6500 Mainz 1

Bagplus Basic Erweiterung für den Apple. 19! neue Befehle: Swap Instring, Inkey, Fillstr, Music Fillpage, Invpage für 49 DM Info anford. Guido Schrörs Corneliusstr. 29, 4154 Tönisvorst 1

Astrologie-Progr. mit Text, Progn. + Partnervergl. Probeausdr. DM 10,– Geb. Dat. Zeit u. Ort Ang. CH. Marquard, Postf. 145, D-7753 Allensbach

* **APPLE IIe, APPLE IIc** * Maschinenschreiben, Gratisinfo, Demo-Diskette 10 DM, Dipl. Ing. M. Decker, Johannastr. 64, 4220 Dinslaken, Tel.: 0 21 34/3 68 83

APPLE-Programme nach Wunsch: Rüter, Rahdener Str. 65, 4955 Hille

Ankauf Software

Suche **Mailbox-Programm für APPLE IIe.** Angebote an A. Krämer, Talstr. 37, 6057 Dietzenbach

Kontakte

Suche Interessierte zur Gründung eines **APPLE-Computerclubs** im Raum 5010 Bergheim zur professionellen Programmerstellung. Tel.: 0 22 71/4 35 62.

Erscheinungstermin für Ausgabe 7/85 ist am 24.6.1985

Ausbildung

Mikroprozessoren Computertechnik

Praxisnahe Fachausbildung durch staatl. geprüften Fernlehrgang. Hard/Software, Maschinensprache, Programmierungstechnik. Computer-Lernsystem wird mitgeliefert. Information kostenlos. ISF-Lehrinstitut. 2800 Bremen 34/Abt. 8-184

Verschiedenes

MC 3470 (der Leseverstärker auf dem Analogboard): DM 10,–, A. Deckers, PF 967, 7 Stuttgart 1

Disketten-Box 49,00 DM für 80 5 1/4" Disketten, M. Ziemer, Belmer Str. 14, 4500 Osnabrück, 05 41/7 43 33 o. 3 78 59

Suche **„Schriftliches"** zu ROBO1ØØØ + APPLEWORKS, Tel. 02 11/33 58 42

Suche Gebrauchsanleitung für **Visiplot** und **Visitrend** Tel.: 0 22 33/7 51 02

Floppysubsystem für Apple II
2 × 3,5" TEAC FD-35F
max. 2 × 1MB
Autopatch-Controller
inkl. Manual und SW
ALU-Profil-Gehäuse m. Netzt.
komplett anschlußfertig
nur DM 2398,–

Telekommunikation
Anschlußkabel
Dataphon S21d
Terminalsoftware
komplett anschlußfertig
nur DM 398,–

TISCH & ZETTL GdbR
Elektronicvertrieb
Rosenstr. 33, 8034 Germering
Telefon 089/841 68 17

Einkaufsführer

RUNOW Büroelektronik
Keithstr. 26 · 1 Berlin 30 · ☎ 0 30-26 111 26

RUNOW Büroelektronik GmbH
Bachstr. 104 · 2 HH 76 · ☎ 0 40-220 11 55

Für weitere Informationen zu einem der in dieser Ausgabe vorgestellten Produkte stehen Ihnen die Produktkarten zur Verfügung

Bitte verwenden Sie für Kleinanzeigen die vorgedruckten Antwortkarten in diesem Heft.

```
2030 IF P = 2 THEN POKE - 16299,0
2040 POKE - 16297,0
2050 POKE - 16304,0
2060 POKE CO,0: REM jetzt drucken
2070 POKE HI,D: REM eventuell Doppel-Hires
2080 & : REM SUPERDUMP aufrufen
2090 TEXT
2100 HOME : VTAB (22)
2110 PRINT "Nochmal? (J/N)";: GOSUB 2150
2120 IF A$ = "J" GOTO 1570
2130 IF A$ = "N" THEN HOME : END
2140 GOTO 2100
2150 GET A$: PRINT A$;
2160 IF A$ > = "a" THEN A$ = CHR$ ( ASC (A$) - 32)
2170 RETURN
2180 REM
2190 POKE KA,P1
2200 POKE KA + 1,P2
2210 POKE AD + 1,P * 32
2220 POKE RF,1
2230 POKE RI,6
2240 POKE RI + 1,39
2250 IF D = 1 THEN POKE RI + 1,79
2260 POKE BO,191
2270 POKE XF,1
2280 POKE YF,1
2290 POKE HI,D
2300 POKE DP,1
2310 POKE NE,NG
2320 IF D = 1 THEN C$ = STR$ (DG): GOSUB 3280: GOSUB 3510
2330 RETURN
2340 REM
2350 PRINT "Welches Kommando: ";
2360 GET C$: IF (C$ = " ") OR (C$ = CHR$ (13)) THEN C$ =
     "1"
2370 IF VAL (C$) = 0 THEN C$ = "1"
2380 RETURN
2390 REM
2400 HOME
2410 PRINT "<1> 1-fach Druck": PRINT
2420 PRINT "<2> 2-fach Druck": PRINT
2430 PRINT "<3> 4-fach Druck": PRINT
2440 GOSUB 2350
2450 POKE IM, VAL (C$) - 1
2460 RETURN
2470 REM
2480 HOME
2490 PRINT "<1> Positiv": PRINT
2500 PRINT "<2> Negativ": PRINT
2510 GOSUB 2350
2520 IF C$ = "2" THEN POKE XO,255
2530 RETURN
2540 REM
2550 HOME
2560 PRINT "<1> Horizontal": PRINT
2570 PRINT "<2> Vertikal": PRINT
2580 GOSUB 2350
2590 POKE DI, VAL (C$) - 1
2600 RETURN
2610 REM
2620 HOME
2630 PRINT "<1> Kein Rahmen": PRINT
2640 PRINT "<2> Rahmen": PRINT
2650 GOSUB 2350
2660 POKE FR, VAL (C$) - 1
2670 RETURN
2680 REM
2690 HOME
2700 PRINT "<1> Seite 1": PRINT
2710 PRINT "<2> Seite 2": PRINT
2720 GOSUB 2350
2730 POKE AD + 1, VAL (C$) * 32
2740 RETURN
2750 REM
2760 HOME
2770 PRINT "<1> Automatisches LF nach CR": PRINT
2780 PRINT "<2> Kein automatisches LF nach CR": PRINT
2790 GOSUB 2350
2800 POKE AL, VAL (C$) - 1
2810 RETURN
2820 REM
2830 HOME
2840 PRINT "<1> Unterste Drucknadel Bit 0": PRINT
2850 PRINT "<2> Unterste Drucknadel Bit 7": PRINT
2860 GOSUB 2350
2870 POKE NE, VAL (C$) - 1
2880 RETURN
2890 REM
2900 HOME
2910 INPUT "Wiederholungsfaktor : ";I
2920 I = ABS (I): IF I > 255 THEN I = 1
2930 IF I = 0 THEN I = 1
2940 POKE RF,I
2950 INPUT "Wiederholungsdistanz: ";I
2960 I = ABS (I): IF I > 255 THEN I = 0
2970 POKE RD,I
2980 INPUT "Linker Rand          : ";I
2990 I = ABS (I): IF I > 559 THEN I = 0
3000 POKE LE + 1, INT (I / 7)
3010 POKE LE,I - INT (I / 7) * 7
3020 INPUT "Rechter Rand         : ";I
3030 I = ABS (I): IF I > 559 THEN I = 279
3040 POKE RI + 1, INT (I / 7)
3050 POKE RI,I -  INT (I / 7) * 7
3060 INPUT "Oberer Rand          : ";I
3070 I = ABS (I): IF I > 191 THEN I = 0
3080 POKE TP,I
3090 INPUT "Unterer Rand         : ";I
3100 I = ABS (I): IF I > 191 THEN I = 191
3110 POKE BO,I
3120 INPUT "X-Faktor             : ";I
3130 I = ABS (I): IF I > 31 THEN I = 1
3140 IF I = 0 THEN I = 1
3150 POKE XF,I
3160 INPUT "Y-Faktor             : ";I
3170 I = ABS (I): IF I > 31 THEN I = 1
3180 IF I = 0 THEN I = 1
3190 POKE YF,I
3200 INPUT "Anzahl der Bilder    : ";I
3210 I = ABS (I): IF I > 8 THEN I = 1
3220 IF I = 0 THEN I = 1
3230 POKE DP,I
3240 INPUT "Kontrollblocknummer : ";I
3250 I = ABS (I): IF I > 255 THEN I = 1
3260 POKE KB,I
3270 REM
3280 IF PEEK (DI) = 0 THEN BT = PEEK (XF) * ( PEEK (RI) +
     PEEK (RI + 1) * 7 - PEEK (LE) - PEEK (LE + 1) * 7 + 1)
3290 IF PEEK (DI) = 1 THEN BT = PEEK (YF) * ( PEEK (BO) +
     PEEK (BO + 1) * 7 - PEEK (TP) - PEEK (TP + 1) * 7 + 1)
3300 BT = (BT + PEEK (RD)) * PEEK (RF) * PEEK (DP)
3310 BH = INT (BT / 256)
3320 BL = BT - INT (BT / 256) * 256
3330 BT$ = STR$ (BT)
3340 L = LEN (BT$)
3350 IF L < 4 THEN FOR I = L TO 3:BT$ = "0" + BT$: NEXT
3360 RETURN
3370 REM
3380 HOME
3390 PRINT "Drucker: ";PF$: PRINT
3400 PRINT "<1>  576 Punkte/Zeile"
3410 PRINT "<2>  640 Punkte/Zeile"
3420 PRINT "<3>  768 Punkte/Zeile"
3430 PRINT "<4> 1152 Punkte/Zeile"
3440 PRINT "<5> 1280 Punkte/Zeile"
3450 PRINT "<6>  856 Punkte/Zeile"
3460 PRINT "<7>  960 Punkte/Zeile"
3470 PRINT "<8> 1088 Punkte/Zeile"
3480 PRINT
3490 GOSUB 2350
3500 REM *** hier Grafiksteuerzeichen POKEn ***
3510 I = PB +  PEEK (KB) * 64 + 16: REM Adresse der
     Grafiksteuerzeichen
3520 ON VAL (C$) GOTO
     3530,3540,3550,3560,3570,3580,3590,3600
3530 C$ = "n": GOTO 3610
3540 C$ = "N": GOTO 3610
3550 C$ = "E": GOTO 3610
3560 C$ = "p": GOTO 3610
3570 C$ = "P": GOTO 3610
3580 C$ = "e": GOTO 3610
3590 C$ = "q": GOTO 3610
3600 C$ = "Q"
3610 POKE I + 2, ASC (C$)
3620 FOR J = 5 TO 8: POKE I + J, ASC ( MID$ (BT$,J - 4,1)):
     NEXT : RETURN
```

SUPERDUMP.EPSON

Die folgenden Zeilen müssen im Listing SUPERDUMP.IMAGEWRITER geändert werden.

```
1000 PF$ = "EPSON": REM Parameter-File
1010 DG = 2: REM Grundeinstellung für Double-Hires
1020 NG = 0: REM Grundeinstellung für Nadel
```

```
3400 PRINT "<1>  480 Punkte/Zeile"
3410 PRINT "<2>  960 Punkte/Zeile"
3420 PRINT "<3>  960 Punkte/Zeile"
3430 PRINT "<4> 1920 Punkte/Zeile"
3440 PRINT "<5>  640 Punkte/Zeile"
3450 PRINT "<6>  576 Punkte/Zeile"
3460 PRINT "<7>  720 Punkte/Zeile"
3470 PRINT
3480 GOSUB 2350
3490 REM *** hier Grafiksteuerzeichen POKEn ***
3500 I = PB + PEEK (KB) * 64 + 16: REM Adresse der
     Grafiksteuerzeichen
3510 ON VAL (C$) GOTO 3520,3530,3540,3550,3560,3570,3580
3520 C$ = "K": GOTO 3590
3530 C$ = "L": GOTO 3590
3540 C$ = "Y": GOTO 3590
3550 C$ = "Z": GOTO 3590
3560 C = 4: GOTO 3600
3570 C = 5: GOTO 3600
3580 C = 6: GOTO 3600
3590 POKE I,4: POKE I + 2, ASC (C$): POKE I + 3,BL: POKE
     I + 4,BH: RETURN
3600 POKE I,5: POKE I + 2, ASC ("*"): POKE I + 3,C: POKE
     I + 4,BL: POKE I + 5,BH: RETURN
```

Die Zeilen 3610 und 3620 entfallen.

Tabelle 1: Übergabeparameter

```
$0300 - Kontrollblock-Nummer: Nummer des aktuellen Kontroll-
        blocks. Erklärung im Text.
        (siehe Tabelle 2)
$0302 - Adresse des Kontrollblockes: Startadresse des
        Kontrollblocks #0 im üblichen Low/high-Format.
$0304 - Druckschwärze: (Wertebereich 1..3)
        1 = 1-fach Druck
        2 = 2-fach Druck
        3 = 4-fach Druck
$0306 - XOR-Byte: (Wertebereich 0, 255)
        $00 = schwarz auf weiß
        $FF = weiß auf schwarz
$0308 - Richtung: (Werteberich 0, 1)
        0 = horizontal
        1 = vertikal
$030A - Rahmen: (Wertebereich 0, 1)
        0 = kein Rahmen um das Bild zeichen
        1 = Rahmen zeichnen (z.B. zum Ausschneiden)
$030C - Bildadresse: (Wertebereich 0..2↑16-1)
        im Low/high-Format
        $2000 für Seite 1
        $4000 für Seite 2
        oder eine beliebige andere Adresse im RAM-Bereich
$030E - Wiederholungsfaktor: (Wertebereich 0..255)
        Anzahl der Wiederholungen des gleichen Bildes
        nebeneinander
$0310 - Wiederholungs-Abstand: (Wertebereich 0..255)
        Anzahl der Bits, die zwischen zwei oder mehr
        aufeinanderfolgenden Bilder frei bleiben soll.
$0312 - linker Bildrand: (Wertebereich 0..559)
        Folgendes Format muß eingehalten werden:
        $0312 = Linker Rand mod 7
        $0313 = Linker Rand div 7
        Bsp: Linker Rand = 10 daraus folgt
        $0312 = 3, $0313 = 1 (weil 10 = 1 * 7 + 3)
$0314 - rechter Bildrand: siehe linker Bildrand
$0316 - oberer Bildrand: (Wertebereich 0..191)
$0318 - unterer Bildrand: (Wertebereich 0..191)
$031A - Xfaktor: (Wertebereich 0..31)
        Dehnung des Bildes in X-Richtung
$031C - Yfaktor: siehe Xfaktor
$031E - Hires 560: (Wertebereich 0, 1)
        0 = Drucken der normalen 280*192-Grafik
        1 = Drucken der Apple-IIe- oder Apple-IIc-
        560*192-Grafik
$0320 - Auto-LF: (Wertebereich 0, 1)
        0 = sende LF nach CR
        1 = sende kein LF nach CR
$0322 - Anzahl der verschiedenen Teilbilder:
        (Wertebereich 1..8)
        Bsp: 1. Teilbild (linke Hälfte) Seite 1,
             2. Teilbild (rechte Hälfte) Seite 2,
        dann setzt man:
        $30C:00 20 (Adresse des 1. Teilbild)
        $322:02    (2 Teilbilder)
        Das heißt: Das 2. Teilbild wird bei $4000 beginnen,
        denn es werden pro Teilbild $2000 zur Anfangs-
        adresse aufaddiert.
$0324 - Drucknadel-Richtung: (Wertebereich 0, 1)
        Für Epson Drucker = 0,
        für Imagewriter = 1,
        für alle anderen Drucker selbst ausprobieren
$0326 - Kommando-Byte
        0 = Drucke Bild
        1 = Verschiebe von Auxiliary-RAM $2000 nach
            Main-RAM $4000
        2 = Verschiebe von Main-RAM $4000 nach
            Auxiliary-RAM $2000
        3 = Konvertiere ein Double-Hires-Bild in 2 normale
            Hires-Bilder mit
             linker Hälfte in Seite 1 ($2000) und
             rechter Hälfte in Seite 2 ($4000)
        4 = Konvertiere 2 normale Hires-Bilder von
             Seite 1 (linke Hälfte) und
             Seite 2 (rechte Hälfte)
            in ein Double-Hires-Bild.
```

Die obige Grafik wurde von Martin Schmidt erstellt und mit Superdump ausgedruckt.

Tabelle 2: Print-File

Epson

```
Block #0: für Horizontaldruck

Pinit: 9000- 03 1B 33 17 00 00 00 00 Länge 3: ESC,
       9008- 00 00 00 00 00 00 00 00 "3", CHR$(23)
Pline: 9010- 04 1B 4B 18 01 00 00 00 Länge 4: ESC,
       9018- 00 00 00 00 00 00 00 00 "K", CHR$(24), CHR$(1)
Plfl:  9020- 03 1B 4A 01 00 00 00 00 Länge 3: ESC,
       9028- 00 00 00 00 00 00 00 00 "J", CHR$(1)
Pexit: 9030- 02 1B 32 00 00 00 00 00 Länge 2: ESC,
       9038- 00 00 00 00 00 00 00 00 "2"


Block #1: für Vertikaldruck

Pinit: 9040- 03 1B 33 14 00 00 00 00 Länge 3: ESC,
       9048- 00 00 00 00 00 00 00 00 "3", CHR$(20)
Pline: 9050- 04 1B 4B C0 00 00 00 00 Länge 4: ESC,
       9058- 00 00 00 00 00 00 00 00 "K", CHR$(192), CHR$(0)
Plfl:  9060- 03 1B 4A 01 00 00 00 00 Länge 3: ESC,
       9068- 00 00 00 00 00 00 00 00 "J", CHR$(1)
Pexit: 9070- 02 1B 32 00 00 00 00 00 Länge 2: ESC,
       9078- 00 00 00 00 00 00 00 00 "2"
```

Imagewriter

```
Block #0: für Horizontaldruck

Pinit: 9000- 06 1B 54 31 36 1B 3E 00 Länge 6: ESC,
       9008- 00 00 00 00 00 00 00 00 "T16", ESC, ">"
Pline: 9010- 08 1B 6E 1B 47 30 32 38 Länge 8: ESC,
       9018- 30 00 00 00 00 00 00 00 "n", ESC, "G0280"
Plfl:  9020- 00 00 00 00 00 00 00 00 Länge 0
       9028- 00 00 00 00 00 00 00 00
Pexit: 9030- 04 1B 3C 1B 41 00 00 00 Länge 4: ESC,
       9038- 00 00 00 00 00 00 00 00 "<", ESC, "A"
```

```
Block #1: für Vertikaldruck

Pinit: 9Ø4Ø- Ø6 1B 54 31 34 1B 3E ØØ Länge 6: ESC,
       9Ø48- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ "T14", ESC, ">"
Pline: 9Ø5Ø- Ø8 1B 6E 1B 47 3Ø 31 39 Länge 8: ESC,
       9Ø58- 32 ØØ ØØ ØØ ØØ ØØ ØØ ØØ "n", ESC, "GØ192"
Plfl:  9Ø6Ø- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ Länge Ø
       9Ø68- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ
Pexit: 9Ø7Ø- Ø4 1B 3C 1B 41 ØØ ØØ ØØ Länge 4, ESC,
       9Ø78- ØØ ØØ ØØ ØØ ØØ ØØ ØØ ØØ "<", ESC, "A"
```

Tabelle 3

Frage	Antwort
zu Bild 1:	
Apple IIe Double-Hires-Grafik?	"N"
Bild speichern?	"N"
Grafik schon im Speicher?	"N"
Welche Grafik laden?	<Name des Grafikfiles angeben>
Ist dies das richtige Bild?	"J"
Grundeinstellung?	"J"
zu Bild 2:	
Anfang wie Bild 1	
Grundeinstellung?	"N"
1-fach Druck	"1"
Positiv	"1"
Horizontal	"1"
Kein Rahmen	"1"
Seite 1	"1"
Automatisches LF nach CR	"1" für Epson und Imagwriter
Unterste Drucknadel Bit Ø	"1" für Epson "2" für Imagewriter
Wiederholungsfaktor	"1"
Wiederholungsdistanz	"Ø"
Linker Rand	"Ø"
Rechter Rand	"279"
Oberer Rand	"Ø"
Unterer Rand	"191"
X-Faktor	"1"
Y-Faktor	"1"
Anzahl der Bilder	"2"
Kontrollblock Nummer	"Ø"
Grafikmodi	"2" für Epson "4" für Imagewriter

Anmerkung zu Bild 2: Dies sind zwei normale 28Ø*192-Punkte-Bilder, die nebeneinander im Speicher liegen, aber zusammen gedruckt werden. Dies kann beim Drucken für Graf-quattro-Bilder benutzt werden.

zu Bild 3:

Apple IIe Double-Hires-Grafik? "J"
sonst wie Bild 1

ccp datentechnik

Neu Neu Neu Neu Neu

640 KByte-Drives für den Apple //c!!

- 5¼- od. 3½-Zoll-Format (Teac FD55/35-F)
- FD55-F umschaltbar auf 35/40 Track
- Anschluß an die externe Laufwerkbuchse
- Kein Öffnen des Gerätes erforderlich
- Einfache Anpassung für DOS 3.3, UCSD-Pascal und PRODOS durch menügeführten Patch
- Anpassung von CP/M in Verbindung mit einer Z 80-Zusatzplatine in Vorbereitung
- anschlußfertig im Gehäuse . . . . . **DM 990,–**

Festplatten für Apple II (//e)

- 5¼ Zoll-Format (Slimline)
- Booten direkt von der Festplatte in DOS 3.3, UCSD-Pascal, PRODOS und CP/M 2.2 / 3.0
- Gemischtbetr. mit 35/40/80/160 Track-Drives
- Copy- und Install-Programme im Lieferumfang
- Umfangreiches Manual
- z. B. 12 MB form. incl. Netzteil u. Contr., anschlußfertig an Ihren Apple . . . . . **DM 4552,–**

640 KByte-Drives für Apple II (//e)

- 5¼- od. 3½-Zoll-Format (Teac FD55/35-F)
- FD55-F umschaltbar auf 40 Track (Apple kompatigel)
- Installationssoftware für DOS 3.3, UCSD-Pascal, CP/M 2.2, CP/M 2.23 (60K), PRODOS, AP22, ALS CP/M+
- Umfangreiches Handbuch
- Anschlußfertige Auslieferung incl. Contr. und 2 Drives
- Diskstation 55II (2 Teac FD55-F, 1.2 MB) . . . **DM 1649,–**
- Diskstation 35II (2 Teac FD35-F, 1.2 MB) . . . **DM 1540,–**

Alles für Ihren Apple

Info bei:

ccp-datentechnik

Herderstraße 12 – 2000 Hamburg 76
Telefon 040/225676

INTUS – Lernprogramme

in deutscher Sprache für Apple IIe und IIc.

- Rechtschreib-Trainer. 4 Übungsarten, 16 Bereiche, über 4000 Wörter und Kurzsätze. Sehr wirkungsvoll, gute Erfolge bei Legasthenie DM 125,–
- Wortschatztrainer Englisch und Französisch, je DM 98,–
- Maschineschreiben wie der Blitz. 20 Lektionen. DM 188,–
- Basic-Lernprogramm. 14 Lektionen. DM 295,–
- Lesen wie der Blitz.Wirkungsvolles Lesetraining. DM 98,–
- Kinderschule. Beliebte Vorschulprogramme. DM 69,–
- Demo-Disk. mit Kostproben aus 9 Programmen. DM 10,–

Katalog gratis.

INTUS SOFTWARE

Kaiserstraße 21 · 7890 Waldshut-Tiengen · Telefon 07751-7920

Macintosh

512 K

Speichererweiterung

Professionelle Umrüstung in 24 Std.
Zum Superpreis von nur DM **1680,--**
Schappach Computer 68 Mannheim
S6,36 Telefon 0621-12662

BUCH-SHOP

Betriebssystem CP/M

Vom Monitorprogramm zum Mehrbenutzersystem.
Von Jürgen Plate.
1984, 351 Seiten, 30 Abb., 3 Tab., geb., DM 56,–

Das Buch beschreibt ausführlich die Kommandos, ihre genaue Syntax und die einzelnen Teilprogramme von CP/M wie BIOS (systemspezifischer Teil), ED (Editor), ASM (Asembler, inklusive einer Beschreibung des 8080-Befehlssatzes), SYSGEN und STAT.
Der Beschreibung von CP/M ist das Listing eines komfortablen Monitorprogramms für Z-80-Computer vorangestellt, das eine elementare Programmierung auf Maschinenebene erlaubt, solange man CP/M noch nicht geladen hat. Das kann z. B. zur Fehlersuche sehr nützlich sein. Am Schluß des Buches findet sich auch eine Kurzbeschreibung der Multitasking-/Multiuser-Betriebssysteme.

Das Buch zum Apple II

von Erich Esders
1985, 210 S., 119 Abb., geb, DM 54,–

Wenn hier vom Apple II gesprochen wird, so gilt das auch für den IIplus, den IIeuroplus und die IIe-Versionen sowie für den ganzen „Apple-Nachbau". Das Buch ist ein Wegweiser durch diesen Rechner, um mit ihm schneller und effektiver zu arbeiten. Es geht hier weniger um das elementare Programmieren des Rechners, sondern um Assemblerprogramme, die extensiv Monitor-ROM-Subroutinen benutzen. Diese hat der Autor nach Sachgebieten geordnet, z. B. Mathematik, Graphik, String-Bearbeitung + Disassembler-Listings und diese wiederum mit Erklärungen und Applikationen komplettiert. Eine ausreichende Dokumentation ist dabei immer gewährleistet. Sie geht schrittweise vor, von der Aufgabenstellung über die Programmentwicklung bis zum lauffähigen Maschinenprogramm. Die angebotenen Beispiele sind ausbaufähig und lassen der eigenen Kreativität reichlichen Spielraum. Viele neuartige Tips und Tricks wird auch der beschlagene Apple-Benutzer begrüßen.
Aus dem Inhalt:
Der Mikroprozessor des APPLE II. Der APPLE II und seine Speicheraufteilung. APPLESOFT und seine Arbeitsspeicher-Bereiche. Der MICROSOFT-Basic-Interpreter: Die Zeichen-Lese-Routine. Interpretierer und Lokalisierer. Handler-Routinen. BASIC/Maschinensprache-Interfaces. DISAS-Generator. Unterprogramme im APPLESOFT-Basic-Interpreter: Softschalter und -Flags. Ausdrucks-Interpretierer. Low-Resolution-Graphik. Fehler-Behandlung. Applikationen: Arithmetik-Demonstration „FP-CALC". Hex-Dumps der Applikationen. BASIC-Monitor BASMON/D: Vorstellung der neuen Kommandos. Das Programm „BASMON/D". Implementierung und Laufbeispiele. BASIC-Interpreter-Vergleich APPLE II – Commodore 64: Arithmetik-Demonstration „FP-CALC/64". Listen: Die Token des APPLESOFT-Basic.

Apple II ROM Listing

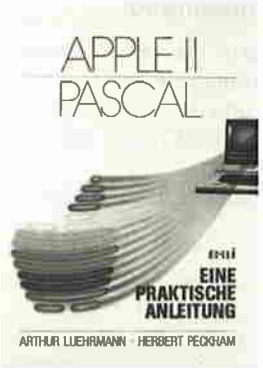
von Matthias Buck
1984, 116 S., Kart., DM 59,–

Das deutsche Apple-II-ROM-Listing ist da!
Einleitung zum prinzipiellen Ablauf des Applesoftinterpreters:
- Aufbau und Verarbeitung der/des Programmtextes – Variablentabelle – String-space – Fließkommaformate – Basicstacks (GDSUB, FOR-NEXT, ...)
- Beschreibung der wichtigsten Unterprogramme, z. B. Variablensuche, Garbage collection, Ausdrucksauswertung, CHRGET, ...
- Vollständig disassemblierte und sehr ausführlich deutsch kommentierte Auflistung des Applesoft-BASIC-Interpreters
- Übersichtliche Auflistung aller vom Interpreter benutzten RAM-Zeilen mit allen Verwendungszwecken
- Über 150 ausführlich dokumentierte Unterprogramme:
  – Funktion
  – Ein/Ausgabeparameter

Auch für Apple-IIe und c und Kompatible!

Apple II Pascal

Eine praktische Anleitung
von Arthur Luehrmann und Herbert Peckham
1982, 544 S., kart., DM 59,–



Dieses Buch ist unentbehrlich für alle, die die Programmiersprache PASCAL lernen wollen und Zugang zu einem Apple Computer haben.
Sie benötigen keinerlei Vorkenntnisse, sondern lernen an Hand von Beispielen und Übungen, wie man selbst PASCAL-Programme entwikkelt und sie austestet und werden allmählich von Kapitel zu Kapitel vertrauter im Umgang mit dem Apple Computer.

Start mit Apple-Logo für II, IIe und IIc

Das kleine Logo-Einmaleins: Grafik * Text * Musik
Von D. Senftleben
1985, 222 Seiten, DM 35,–

Viele Mikrocomputer-Hersteller bieten für ihre Geräte neben BASIC und anderen Programmiersprachen zunehmend auch Logo an. Durch ihre Benutzerfreundlichkeit hat diese Sprache bereits viele Freunde im Ausbildungs- und Freizeitbereich gefunden. Dabei ist Logo eine mächtige Sprache, die auch dem anspruchsvollen Anwender kaum Wünsche offenläßt. Mittels Schildkrötengrafik wird das kleine Logo-Einmaleins in 12 Lektionen entwikkelt. Große Bildschirmfotos begleiten den Leser durch die Lektionen. Das Buch verlangt aktive Mitarbeit. Es hat seinen Platz neben dem Computer und gibt Hilfen und Anregungen für eigenes Forschen. Dank des bausteinorientierten Konzepts kann jeder seine eigenen Teilbausteine erzeugen und sie zu neuen Blöcken zusammenfügen. Neben dem Einmaleins werden neue Einsatzbereiche für den Einsteiger erschlossen. Musik und Sound fehlen nicht.
In diesem Buch werden die beiden offiziellen Logo-Produkte der Firma Apple für die Rechnerfamilie Apple II, IIe und IIc behandelt und deren Unterschiede verdeutlicht. Weiterhin sind sämtliche Apple-Logo-Vokabeln übersichtlich zusammengestellt. Dieses Buch ist ideal zum problemlosen und vergnüglichen Start in die Apple-Logo-Welt.

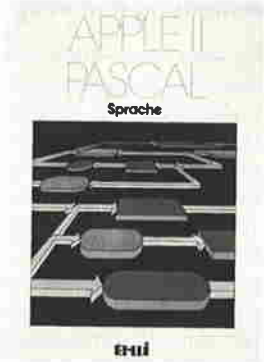
Apple II Anwenderhandbuch

von Lon Poole
1982, 450 S., zahlr. Abb., kart., DM 56,–

Auch für diesen Computer haben wir den richtigen Leitfaden. Er erspart Ihnen zeitraubendes und nutzloses Suchen nach der wirklich verwendbaren Dokumentation für Ihren Computer. Das Anwenderhandbuch beschreibt zum einen den beliebten Apple II-Computer als solchen und gibt zum anderen ausführlich Auskunft über die normalen Peripheriebausteine und Zubehör einschließlich Disk-Laufwerken und Drucker. Mit Hilfe dieses Buches werden Sie Ihren Apple II erfolgreich einsetzen, denn der Informationsgehalt geht weit über das hinaus, was herstellerseitig an Literatur angeboten wird. Sie lernen BASIC auf zwei verschiedene Arten zu verwenden. Wie man den Gebrauch von Klang, Farbe und Grafik zum Optimum führt. Sie erhalten Tips für fortgeschrittene Programmerstellung. Sie erfahren die Verwendung des Maschinensprachen-Monitors u. v. m. Mit dem Apple II-Anwenderhandbuch werden Ihnen alle Möglichkeiten eröffnet, die in diesem Computer stecken.

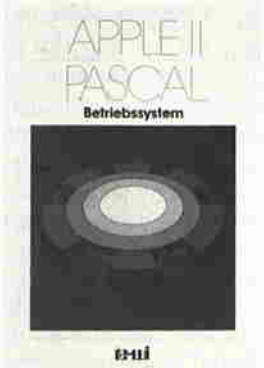
Apple II Pascal

Sprache
1985, 197 S., DM 39,–



Apple II Pascal

Betriebssystem
1985, 256 S., DM 49,–



Ralf Menssen

Hardcopy auf Olympia ESW 100 RO

Ein Typenraddrucker wird grafikfähig

Das Programm **OLYMPIA** gibt eine Hardcopy vom hochauflösenden Grafikbildschirm 1 oder 2 auf dem Typenraddrucker Olympia ESW 100 RO aus. Es ist durch leichte Modifikationen im Prinzip für alle Typenraddrucker verwendbar, die ihre horizontale und vertikale Schrittweite auf 1/60 bzw. 1/96 Zoll einstellen können. Es läuft auf dem Apple II mit 48K und allen kompatiblen Rechnern. Die hier vorgestellte Version beginnt bei Speicherplatz $8000 und sollte durch BSAVE OLYMPIA, A$8000, L$1E8 auf Diskette gespeichert werden; es kann jedoch auch an anderer Stelle neu assembliert werden.

Bedienung des Programms

Die auszudruckende Grafik muß sich im Speicher befinden. Dann lädt man die Druckroutine mit BLOAD OLYMPIA und startet sie mit CALL 32768. Auf dem Bildschirm erscheint jetzt die Grafikseite 1 und darunter die Frage, welche Seite ausgedruckt werden soll. Antworten Sie mit „1" oder „2".
Als nächstes müssen Sie entscheiden, ob das Bild positiv oder negativ erscheinen soll, d.h. ob ein weißer Punkt auf dem Bildschirm auf dem Papier schwarz oder weiß werden soll. Geben Sie „P" oder „N" ein.
Als letztes müssen Sie noch die vertikale und die horizontale Schrittweite in 1/60 bzw. 1/96 Zoll angeben. Als günstige Eingabe hat sich „02" für die horizontale und „03" für die vertikale Schrittweite herausgestellt. Sie erhalten dann eine Hardcopy, die ein quer eingespanntes DIN-A4-Blatt gut ausfüllt. Außerdem decken sich die Punkte dann sehr gut.
Haben Sie diese Eingaben gemacht, brauchen sie etwas Geduld, denn der Typenraddrucker muß jetzt ca. 50000 Positionen ansteuern und wahlweise einen Punkt oder ein Leerzeichen drucken. Aber das Warten lohnt sich, denn die Qualität der Hardcopy übertrifft Matrixdrucker der gleichen Preisklasse klar.

Zur Schonung des Typenrades stellen Sie bitte den geringsten Anschlag am Drucker ein. Außerdem sollte man ein Endlosfarbband verwenden, da ein Karbonband zu schnell verbraucht ist.

Programmaufbau

Es werden die zwei Unterprogramme COUT (Ausgabe eines Zeichens bei $FDED) und RDKEY (Lesen eines Zeichens bei $FD0C) des Apple-Betriebssystems benutzt.
Die druckerspezifische Befehle lauten wie folgt:

ESC P – Auto-Linefeed ein-/ausschalten ($1B $50),
ESC $FF nn – vertikale Schrittweite definieren ($1B $0C nn),
ESC CR nn – horizontale Schrittweite definieren ($1B $0D nn),
SP – Leerschritt ($20),
CR – Wagenrücklauf ($0D),
„." – Punkt ($2E).

Das Programm ist für eine Druckerschnittstelle in Slot 1 ausgelegt. Festgelegt ist dies durch die Labels PORTDATA und HANDSHK. Für Slot 2 müßte PORTDATA zu $C0A0 und HANDSHK zu $C2C1 geändert werden. Für manche Interfaces ist ein anderes Übergabeprotokoll erforderlich, das Sie dem zugehörigen Handbuch entnehmen können.
Für künstlerische Effekte bietet sich die Zeile 179 an. Dort wird festgelegt, daß der Drucker einen Punkt (= $2E) druckt. Als Alternative wären z.B. der Schrägstrich oder das Ausrufezeichen denkbar. Ändert man dann noch die Schrittweiten, ließe sich die Grafik „zoomen". Weitere Möglichkeiten sind Ihrer Phantasie überlassen.

OLYMPIA

```
                  1   ****************************************
                  2   **     HARDCOPY APPLE - ESW1ØØ-RO     **
                  3   **   Von Ralf Menssen   Berlin 1984   **
                  4   ****************************************
                  5   *
                  6            ORG  $8ØØØ
                  7
                  8   COUT     EQU  $FDED      ;Zeichen an Bildschirm
                  9   RDKEY    EQU  $FDØC      ;Zeichen von Tastatur
                  1Ø  INVERT   EQU  $Ø6        ;Bild invertiert ausgeben
                  11  WORKREG  EQU  $Ø7        ;Arbeitsregister
                  12  BASADR   EQU  $1A        ;Bildschirm-Basisadresse
                  13  OUTCOUNT EQU  $1C        ;Zählregister
                  14  MEDCOUNT EQU  $1D        ;Zählregister
                  15  KEYCHR   EQU  $ED        ;Tastenspeicher
                  16  STEPY    EQU  $EE        ;Schrittweite Y
                  17  STEPX    EQU  $EF        ;Schrittweite X
                  18  ZWADR    EQU  $F9        ;Adressen-Übergaberegister
                  19  ENDADR   EQU  $FB        ;endgültige Basisadresse
                  2Ø  INCOUNT  EQU  $FE        ;Zählregister
                  21  ZWERG    EQU  $FF        ;Zwischenergebnis
                  22  PORTDATA EQU  $CØ9Ø      ;Datenport Slot 1
                  23  HANDSHK  EQU  $C1C1      ;Handshakeport (Bit7 = READY)
                  24  *
                  25  ****************************************
8ØØØ: A9 ØØ       26           LDA  #$ØØ
8ØØ2: 8D 5Ø CØ    27           STA  $CØ5Ø      ;HGR1
8ØØ5: 8D 53 CØ    28           STA  $CØ53      ;  +
8ØØ8: 8D 54 CØ    29           STA  $CØ54      ;TEXT
8ØØB: 8D 57 CØ    3Ø           STA  $CØ57      ;einschalten
                  31  ****************************************
                  32  ** PARAMETER EINGEBEN                 **
                  33  ****************************************
8ØØE: A2 ØØ       34           LDX  #$ØØ
8Ø1Ø: BD 8E 81    35   ZUULIE  LDA  DATEN,X    ;"BILDSCHIRM 1 ODER 2"
8Ø13: 2Ø ED FD    36           JSR  COUT       ;
8Ø16: E8          37           INX             ;ausgeben
8Ø17: EØ 2E       38           CPX  #$2E
8Ø19: DØ F5       39           BNE  ZUULIE
                  4Ø  ****************************************
8Ø1B: 2Ø ØC FD    41   FALSCH  JSR  RDKEY
8Ø1E: C9 B1       42           CMP  #$B1
8Ø2Ø: FØ Ø9       43           BEQ  SCREEN1    ;"1" oder "2" von
8Ø22: C9 B2       44           CMP  #$B2       ;Tastatur holen
8Ø24: DØ F5       45           BNE  FALSCH
8Ø26: AØ 4Ø       46           LDY  #$4Ø       ;HGR-Seite 2
8Ø28: 4C 2D 8Ø    47           JMP  WET
8Ø2B: AØ 2Ø       48   SCREEN1 LDY  #$2Ø       ;HGR-Seite 1
8Ø2D: A2 ØØ       49   WET     LDX  #$ØØ
8Ø2F: 86 1A       5Ø           STX  BASADR
8Ø31: 84 1B       51           STY  BASADR+1
                  52  ****************************************
8Ø33: A2 2E       53           LDX  #$2E
8Ø35: BD 8E 81    54   BIGGI   LDA  DATEN,X    ;"POSITIV ODER NEGATIV"
8Ø38: 2Ø ED FD    55           JSR  COUT       ;auf Bildschirm
8Ø3B: E8          56           INX             ;ausgeben
8Ø3C: EØ 4A       57           CPX  #$4A
8Ø3E: DØ F5       58           BNE  BIGGI
                  59  ****************************************
8Ø4Ø: A9 ØØ       6Ø           LDA  #$ØØ
8Ø42: 85 Ø6       61           STA  INVERT
8Ø44: 2Ø ØC FD    62   SLX     JSR  RDKEY
8Ø47: C9 DØ       63           CMP  #$DØ       ;Eingabe = P?
8Ø49: FØ Ø8       64           BEQ  POSITIV
8Ø4B: C9 CE       65           CMP  #$CE       ;Eingabe = N?
8Ø4D: DØ F5       66           BNE  SLX
8Ø4F: A9 Ø1       67           LDA  #$Ø1       ;invertiert
8Ø51: 85 Ø6       68           STA  INVERT     ;drucken
                  69  ****************************************
8Ø53: 2Ø 45 81    7Ø   POSITIV JSR  DATAUS
8Ø56: A9 D8       71           LDA  #$D8
8Ø58: 2Ø 53 81    72           JSR  HOLEN      ;Schrittweite X
8Ø5B: 85 EF       73           STA  STEPX      ;holen
8Ø5D: 2Ø 45 81    74           JSR  DATAUS
8Ø6Ø: A9 D9       75           LDA  #$D9       ;Schrittweite Y
8Ø62: 2Ø 53 81    76           JSR  HOLEN      ;holen
8Ø65: 85 EE       77           STA  STEPY
                  78  ****************************************
                  79  ** INITIALISIERUNG                    **
                  8Ø  ****************************************
8Ø67: A9 ØØ       81           LDA  #$ØØ
8Ø69: 8D 52 CØ    82           STA  $CØ52      ;TEXT ausschalten
8Ø6C: A9 ØD       83           LDA  #$ØD
8Ø6E: 2Ø 3C 81    84           JSR  PRINT      ;CR ausgeben
8Ø71: A5 1B       85           LDA  BASADR+1
```


r+r electronic
Elektronik · Fachliteratur · Personal-Computer

BASF qualimetric®
Disketten 1.Wahl

BASF-Disketten softsektoriert in Pappkarton
5,25"-Disketten alle **ohne Aufpreis** mit Verstärkungsring

Type	Beschreibung	10 Stk	100 Stk
1XV	48 TPI, 1-seitig, einfache Dichte	46,90	459,00
1DV	48 TPI, 1-seitig, doppelte Dichte	49,50	484,50
2/96V	96 TPI, 2-seitig, doppelte Dichte	82,50	799,50

3M Sicherheits-Disketten
5,25"-Disketten **mit Verstärkungsring**, softsektoriert

744-0	48 TPI, 1-seitig, doppelte Dichte	49,50	485,00
745-0	48 TPI, 2-seitig, doppelte Dichte	72,50	679,00
747-0	96 TPI, 2-seitig, doppelte Dichte	89,50	855,00

Disketten – Doubler 14.50
Mit diesem Diskettenlocher können Sie auch die Rückseite Ihrer SS- o. DS-Diskette zum Abspeichern von Daten nutzen.

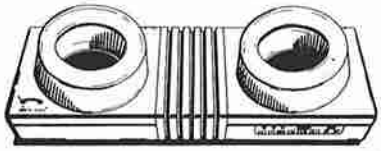
5,25" – Plastikbox 5.95
Formschöne Klappbox aus Hart-PVC für ca. 10 Disketten in den Farben grün, orange, schwarz, hellgrau, elfenbein oder anthrazit. Bitte die gewünschte Farbe angeben. **Mischbar. Staffelpreise: 1 St. 5.95 10 St. 49.50 100 St. 459.-**

5,25" – Disketten-Kästen
Diskettenbox ohne Schloß 24.95
Repräsentative Disketten-Box aus deutscher Fertigung in den Abmessungen (T x B x H) 306x174x144 mm ohne Schloß mit topas farbenem Deckel für **ca. 80 Disketten** und 4 beschriftbaren Stützplatten. **Sehr preiswert!**

Diskettenbox 29.50
aufklappbar
für ca. 50 Disketten
aus stoßfestem, antistatischem transparentem Acryl mit 4 beweglichen, beschriftbaren Stützplatten.

Diskettenbox **abschließbar** 39.50
für ca. 85-100 Disketten
aus antistatischem ABS-Kunststoff, Deckel aus hochwertigem Acryl(rauchglasfarben). Der Trog kannplatzsparend in den Deckel gestellt werden. 9 beweglichen Unterteiler mit beschriftbaren Stützplatten sorgen für Übersicht.

Akustikkoppler s21d 289.-
300 Baud Modem CCITT V.21 Standard



*** Mit FTZ-Nummer * Gebühren- u. anmeldefrei * Anschluß an alle Computer mit V24-Schnittstelle * Vollduplexbetrieb * Answer- und Originate-Modus * MADE IN GERMANY ***

TELETERM 2+/e 198.-
Telekommunikationssoftware für den Apple. * ohne serielles Interface * Datenübertragung über den Gameport * Anschlußfertig an s21d.
Und das bietet Teleterm 2+/e:
Kommunikation über den Bildschirm. Abspeichern empfangener Daten auf Diskette. Daten von Disk laden und senden. Drucken empfangener Daten. Einstellen der Kommunikationsparameter. **Die Software für Kommunikations-Profis.**

Unser großer Hit auch für Ihren Apple //e:

80 Zeichen/64kB-RAM für APPLE //e
dtsch. Qualitätsprodukt
jetzt nur noch... 275.-

Ventilator für Apple 2+/e 99.-
Viele Zusatzkarten entwickeln zusätzliche Wärme und blockieren sogar noch den natürlichen Luftstrom. Der leistungsfähige, geräuscharme Ventilator, sorgt für die notwendige Belüftung.

MASTERCLOCK 325.-
incl. Demodiskette
Universelle Uhrenkarte für Apple //e und //+
Läuft unter DOS, ProDOS und PASCAL.

Z-80 Karte (ohne Sofware) 149.-
ITOH/EPSON-Grafikkarte m.Kabel 230.-
Serielle Karte f. Imagewriter 235.-
RS-232-C-Karte (Seriell) 175.-
Disk-Controller (o.Software) 175.-

Die Aufrüstung Ihres 128k MAC auf 512k kostet bei uns nur: 1590.-

Apple//e, //c + Macintosh Vorführ- und Messegeräte preisgünstig abzugeben.

Ladenverkauf, Vorführung und fachgerechte Beratung

r+r electronic
Elektronik · Fachliteratur · Personal-Computer
6900 Heidelberg 1 Breslauerstr. 29 Tel.06221/781500
Geschäftszeiten: Mo. - Fr. 9-13 +14 - 18 Sa. 9 -13 Uhr
Versandanschrift: 6900 Heidelberg 1 Dammweg 2

```
8073: C9 40      86            CMP  #$40        ;HGR2 angewählt?
8075: D0 05      87            BNE  NEST
8077: A9 00      88            LDA  #$00        ;umschalten
8079: 8D 55 C0   89            STA  $C055       ;auf HGR2
807C: A9 1B      90   NEST     LDA  #$1B        ;ESC ausgeben
807E: 20 3C 81   91            JSR  PRINT
8081: A9 0C      92            LDA  #$0C        ;vertikale
8083: 20 3C 81   93            JSR  PRINT       ;Schrittweite
8086: A5 EE      94            LDA  STEPY       ;ausgeben
8088: 20 3C 81   95            JSR  PRINT
808B: A9 1B      96            LDA  #$1B        ;ESC
808D: 20 3C 81   97            JSR  PRINT
8090: A9 0D      98            LDA  #$0D        ;horizontale
8092: 20 3C 81   99            JSR  PRINT       ;Schrittweite
8095: A5 EF      100           LDA  STEPX       ;ausgeben
8097: 20 3C 81   101           JSR  PRINT
809A: A9 1B      102           LDA  #$1B        ;ESC
809C: 20 3C 81   103           JSR  PRINT
809F: A9 50      104           LDA  #$50        ;AUTO-LINEFEED
80A1: 20 3C 81   105           JSR  PRINT       ;einschalten
                 106  ****************************************
                 107  *********** DRUCKROUTINE ***************
                 108  *                                      *
80A4: A9 03      109           LDA  #$03
80A6: 85 1C      110           STA  OUTCOUNT
80A8: 20 CF 80   111  SCHL1    JSR  SCHL2
80AB: 18         112           CLC
80AC: A5 1A      113           LDA  BASADR
80AE: 69 28      114           ADC  #$28
80B0: 85 1A      115           STA  BASADR
80B2: A5 1B      116           LDA  BASADR+1
80B4: 69 00      117           ADC  #$00
80B6: 85 1B      118           STA  BASADR+1
80B8: C6 1C      119           DEC  OUTCOUNT
80BA: D0 EC      120           BNE  SCHL1
80BC: A9 1B      121           LDA  #$1B        ;AUTO-LINEFEED
80BE: 20 3C 81   122           JSR  PRINT       ;ausschalten
80C1: A9 50      123           LDA  #$50
80C3: 20 3C 81   124           JSR  PRINT
80C6: A9 00      125           LDA  #$00
80C8: 8D 51 C0   126           STA  $C051       ;TEXT wieder
80CB: 8D 54 C0   127           STA  $C054       ;anschalten
80CE: 60         128           RTS              ;Programmende
                 129  ****************************************
                 130  ********** UNTERPROGRAMME **************
                 131  **                                    **
80CF: A6 1A      132  SCHL2    LDX  BASADR
80D1: A4 1B      133           LDY  BASADR+1
80D3: 86 F9      134           STX  ZWADR
80D5: 84 FA      135           STY  ZWADR+1
80D7: A9 08      136           LDA  #$08
80D9: 85 1D      137           STA  MEDCOUNT
80DB: 20 F0 80   138  INLOP    JSR  SCHL3
80DE: 18         139           CLC
80DF: A5 F9      140           LDA  ZWADR
80E1: 69 80      141           ADC  #$80
80E3: 85 F9      142           STA  ZWADR
80E5: A5 FA      143           LDA  ZWADR+1
80E7: 69 00      144           ADC  #$00
80E9: 85 FA      145           STA  ZWADR+1
80EB: C6 1D      146           DEC  MEDCOUNT
80ED: D0 EC      147           BNE  INLOP
80EF: 60         148           RTS
                 149  ****************************************
80F0: A6 F9      150  SCHL3    LDX  ZWADR
80F2: A4 FA      151           LDY  ZWADR+1
80F4: 86 FB      152           STX  ENDADR
80F6: 84 FC      153           STY  ENDADR+1
80F8: A9 08      154           LDA  #$08
80FA: 85 FE      155           STA  INCOUNT
80FC: 20 0B 81   156  LOOP     JSR  SCHL4
80FF: 18         157           CLC
8100: A5 FC      158           LDA  ENDADR+1
8102: 69 04      159           ADC  #$04
8104: 85 FC      160           STA  ENDADR+1
8106: C6 FE      161           DEC  INCOUNT
8108: D0 F2      162           BNE  LOOP
810A: 60         163           RTS
                 164  ****************************************
810B: A0 00      165  SCHL4    LDY  #$00
810D: B1 FB      166  ELP      LDA  (ENDADR),Y ;Byte aus
810F: 85 FF      167           STA  ZWERG       Bildschirmspeicher
8111: A2 07      168           LDX  #$07        ;7 Bildpunkte
8113: A9 00      169  ILP      LDA  #$00
8115: 85 07      170           STA  WORKREG
8117: A5 FF      171           LDA  ZWERG       ;Bildpunkt
8119: 6A         172           ROR              ;holen
811A: 85 FF      173           STA  ZWERG
811C: A5 07      174           LDA  WORKREG
811E: 2A         175           ROL
811F: 45 06      176           EOR  INVERT      ;eventuell
8121: 6A         177           ROR                   invertieren
8122: 90 05      178           BCC  SPC
8124: A9 2E      179           LDA  #$2E        ;Punkt drucken
8126: 4C 2B 81   180           JMP  AUSG
8129: A9 20      181  SPC      LDA  #$20        ;SPACE drucken
812B: 20 3C 81   182  AUSG     JSR  PRINT
812E: CA         183           DEX
812F: D0 E2      184           BNE  ILP
8131: C8         185           INY
8132: C0 28      186           CPY  #$28
8134: D0 D7      187           BNE  ELP
8136: A9 0D      188           LDA  #$0D
8138: 20 3C 81   189           JSR  PRINT       ;Zeile beendet
813B: 60         190           RTS
                 191  ****************************************
813C: 2C C1 C1   192  PRINT    BIT  HANDSHK
813F: 30 FB      193           BMI  PRINT       ;Drucker bereit?
8141: 8D 90 C0   194           STA  PORTDATA    ;Ausgabe SLOT 1
8144: 60         195           RTS
                 196  ****************************************
8145: A2 4A      197  DATAUS    LDX  #$4A
8147: BD 8E 81   198  HAL      LDA  DATEN,X     ;
814A: 20 ED FD   199           JSR  COUT        ;"SCHRITTWEITE"
814D: E8         200           INX              ;auf Bildschirm
814E: E0 5A      201           CPX  #$5A        ;ausgeben
8150: D0 F5      202           BNE  HAL
8152: 60         203           RTS
                 204  ****************************************
8153: 20 ED FD   205  HOLEN    JSR  COUT
8156: A9 BF      206           LDA  #$BF
8158: 20 ED FD   207           JSR  COUT
815B: 20 0C FD   208  WIEDER   JSR  RDKEY
815E: 85 ED      209           STA  KEYCHR
8160: C9 B8      210           CMP  #$B8        ;ist die Eingabe
8162: B0 F7      211           BCS  WIEDER      ;eine Zahl kleiner 9?
8164: C9 AF      212           CMP  #$AF        ;sonst
8166: 90 F3      213           BCC  WIEDER      ;Neueingabe
8168: 85 EE      214           STA  STEPY
816A: 20 ED FD   215           JSR  COUT
816D: 20 0C FD   216  GOZA     JSR  RDKEY
8170: C9 C6      217           CMP  #$C6        ;ASCII der Eingabe
8172: B0 F9      218           BCS  GOZA        ;höher als "F"?
8174: C9 B9      219           CMP  #$B9        ;ist die Eingabe
8176: B0 F5      220           BCS  GOZA        ;eine Zahl?
8178: C9 AF      221           CMP  #$AF
817A: 90 F1      222           BCC  GOZA        ;Steuerzeichen
817C: 20 ED FD   223           JSR  COUT
817F: 29 0F      224           AND  #%00001111  ;rechtes Nibble
8181: 85 EE      225           STA  STEPY
8183: A5 ED      226           LDA  KEYCHR
8185: 29 0F      227           AND  #%00001111  ;rechtes Nibble
8187: 0A         228           ASL              ;ist gleich
8188: 0A         229           ASL              ;linkes Nibble
8189: 0A         230           ASL              ;Schrittweite
818A: 0A         231           ASL
818B: 05 EE      232           ORA  STEPY
818D: 60         233           RTS
                 234  ****************************************
                 235  ************** DATEN ******************
818E: 8D 8D 8D   236  DATEN    HEX  8D8D8D8D8D8D8D8D8D8D8D8D
8191: 8D 8D 8D 8D 8D 8D 8D 8D
8199: 8D
819A: 8D 8D 8D   237           HEX  8D8D8D8D8D8D8D8D8D8D8D
819D: 8D 8D 8D 8D 8D 8D 8D 8D
81A5: 8D
81A6: C2 C9 CC   238           ASC  "BILDSCHIRM 1 ODER 2 ?"
81A9: C4 D3 C3 C8 C9 D2 CD A0
81B1: B1 A0 CF C4 C5 D2 A0
81B1: B1 A0 CF C4 C5 D2 A0 B2
81B9: A0 BF
81BB: 8D 8D 8D   239           HEX  8D8D8D8D
81BE: 8D
81BF: D0 A9 CF   240           ASC  "P)OSITIV ODER N)EGATIV ?"
81C2: D3 C9 D4 C9 D6 A0 CF C4
81CA: C5 D2 A0 CE A9 C5 C7 C1
81D2: D4 C9 D6 A0 BF
81D7: 8D 8D 8D   241           HEX  8D8D8D8D
81DA: 8D
81DB: D3 C3 C8   242           ASC  "SCHRITTWEITE "
81DE: D2 C9 D4 D4 D7 C5 C9 D4
81E6: C5 A0
```

488 Bytes

BUCH-SHOP

Apple DOS 3.3

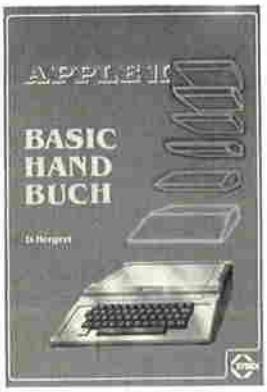
von Ulrich Stiehl
2. Aufl. 1984, 203 S., kart., DM 28,–

Dies ist die erste deutschsprachige Darstellung des Diskettenbetriebssystems DOS 3.3 für den Apple II/II Plus/IIe, die sich sowohl an Applesoft- als auch an Assembler-Programmierer wendet. Sinngemäß ist das Buch zweigeteilt:
Der erste Teil behandelt ausführlich die dem Applesoft-Programmierer zur Verfügung stehenden DOS-Befehle, wobei die Textfiles wegen ihrer großen Bedeutung und der vergleichsweise komplizierten Handhabung besonders dargestellt werden. Viele Textfile-Tricks werden hier zum erstenmal geschildert.
Aber auch im zweiten Teil findet der reine Applesoft-Programmierer insbesondere in dem Kapitel „Vermischte Tips, Tricks und Patches" zahlreiche Anregungen. Im übrigen ist der zweite Teil für Assembler-Programmierer gedacht. Neben einer detaillierten Beschreibung der DOS-Interna enthält dieser Teil elf vollständige RWTS-Anwenderprogramme – z. B. CPM-Refiner, DOS-lose Datendisk, TSL-Maker, File-Reader, Pseudo-Disk-Driver und Fastbrun-Routine –, die Techniken enthüllen, die bislang noch niemals publiziert worden sind. Dieses DOS-Buch ist deshalb der unentbehrliche Begleiter für jeden Apple-Programmierer.

Apple II Basic Handbuch

von Douglas Hergert
304 Seiten, 116 Abb.
DM 32,–

Das Buch ist als Nachschlagewerk konzipiert, daß seinen Platz neben jedem APPLE II, II+ und IIe haben sollte. Es richtet sich an Anfänger und fortgeschrittene Programmierer.



Aus der Praxis heraus präsentiert der Autor Tips und Vorschläge, die das Programmieren leichter und zugleich effizienter machen. Alle Applesoft- und Integer-BASIC-Begriffe sind alphabetisch aufgelistet und werden eingehend erklärt. Dazu werden alle DOS-Befehle (neben vielen Begriffen der Computerterminologie) vorgestellt.
Beispielprogramme zeigen dem Nutzer, wie jeder Befehl funktioniert und helfen, die richtige Anwendung zu üben. Unter anderem lernt der Leser den besten Weg, um FOR/NEXT-Schleifen und IF/THEN-Entscheidungen für seine Zwecke einzusetzen.
Durch die präzise und leicht verständliche Sprache des Autors werden auch schwierige Befehle einfach in der Anwendung.

Apple Maschinensprache

von Don und Kurt Inman
1984, 208 S., zahlr. Abb. und Tabellen, DM 49,–

Dieses Buch ist wahrscheinlich die beste Einführung in die 6502-Programmierung für denjenigen Assembler-Anfänger, der zuvor noch nie ein Maschinenprogramm geschrieben hat.
Aus dem Inhalt:
Applesoft II BASIC – kurzgefaßt – Alles über Zeichen – Alles über Speicher – Alles über Maschinenbefehle – Maschinenprogramme mit BASIC eingeben – Graphik – Text – Ton – Arithmetik – Was tun mit den Maschinenprogrammen

Apple II leicht gemacht

von Joseph Kascmer
1984, 185 S., zahlr. Abb., kart., DM 28,–

Dies ist ein Buch, wie es sich jeder Apple-Anfänger nur wünschen kann: Schrittweise, leichtverständliche Anleitung zum Umgang mit dem Apple mit einigen durchsichtigen, unkomplizierten Beispielen in Applesoft, die ihn nicht Abschrekken, sondern ermutigen sollen, sich mit dem Gerät näher vertraut zu machen. Damit ist „Apple II leicht gemacht" das ideale Einsteigerbuch für den reinen Anwender, der nicht nur „auf den Knopf drücken", sondern zumindest einige Details aus der Black Box namens Apple erfahren will.

Aus dem Inhalt:
Kontrolle des Geräts – Schreiben und Zeichnen auf dem Bildschirm – Geheimnisvolle Abläufe: Programme – Verschiedene Eingriffsmöglichkeiten – Mobile Speicher: Disketten – Kontrollmöglichkeiten – Das Innenleben

Apple Assembler

Tips und Tricks
von Ulrich Stiehl
1984, 226 S., 3 Abb., kart., DM 34,–

„Apple Assembler" wendet sich an alle, die bereits Anfängerkenntnisse der 6502-Programmierung haben – z. B. aufgrund des Buches „Apple Maschinensprache" – und nunmehr ein Nachschlagewerk für ihren Apple II Plus/IIe/IIc suchen, in dem alle wichtigen ROM-Routinen sowie eine Vielzahl sonstiger Hilfsprogramme in einer systematischen Form zusammengestellt werden. Insgesamt umfaßt dieses Buch über 40 Utilities, darunter mehrere völlig neuartige Programme wie Double-Lores, Double Hires, Screen-Format u. a.
Der erste Teil enthält ein Repetitorium der wichtigsten Befehle, Adressierungsarten und sonstigen Besonderheiten des 6502.
Im zweiten Teil werden alle Adressen des Monitors zusammengestellt, die für Assembler-Programmierer von Nutzen sein können. Darüber hinaus findet der Leser Unterroutinen für hexadezimale Addition/Subtraktion/Multiplikation/Division, Binär-Hex-ASCII-Umwandlung usw.
Der dritte Teil befaßt sich mit der Speicherverwaltung der Language Card und der IIe-64K-Karte und enthält Move-Programme zum Verschieben von Daten in die und aus der Language Card sowie der 64K-Karte.
Der vierte Teil ist dem Applesoft-ROM gewidmet und listet eine große Anzahl nützlicher Interpreter-Adressen. Bei den Utility-Programmen liegt das Schwergewicht auf Fließkommamathematik einschließlich Print Using.
Der letzte Teil behandelt den Text- und Graphikspeicher. Neben einem professionellen Maskengeneratorprogramm werden auch Routinen zur Double-Lores- und Double-Hires-Grafik vorgestellt.

Arbeiten mit dem Macintosh

von N. Hesselmann
416 Seiten, 320 Abb. DM 54,–

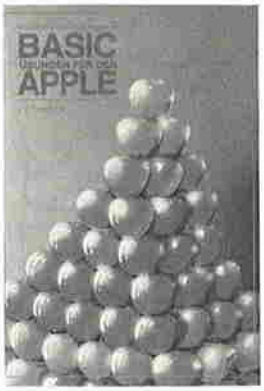
Das Buch erklärt den Umgang mit dem Macintosh von Grund auf, wobei auch auf elementare Dinge eingegangen wird, wie z. B. die Benutzung der Tastatur und der Maus, das Einlegen von Disketten und den Systemstart. Ganz besonderes Augenmerk wird auf die Erklärung der speziellen Software-Umgebung des Macintosh gelegt, wobei das Menü- und Fensterkonzept sowie das Anwählen durch Piktogramme gekennzeichneter Funktionen klar dargestellt wird.

Der Umgang mit den Programmen MacPaint und MacWrite wird erläutert; dies geschieht teilweise anhand von Beispielen, die leicht nachvollzogen werden können. Ein umfangreiches Kapitel ist dem für den Macintosh erhältlichen Microsoft-BASIC gewidmet.

BASIC Übungen für den Apple

von J. P. Lamoitier
1983, 252 S., zahlr. Abb., kart., DM 38,–

Das Buch ist konzipiert, allen Apple-Anwendern Applesoft-BASIC durch praktische Übungen an Hand von reellen Programmen beizubringen. Datenverarbeitung, Statistik, kommerzielle Programme, Spiele und vieles mehr. Jede Übung beinhaltet eine Beschreibung der Problemstellung, eine Analyse der Lösungsmöglichkeiten, ein Flußdiagramm und ein fertiges Programm samt Probelauf.



Aus dem Inhalt:
Ihr erstes BASIC-Programm – Flußdiagramme – Übungen mit Integerzahlen – Elementare Beispiele aus der Geometrie – Allgemeine Übungen aus der Datenverarbeitung – Mathematische Berechnungen – Kaufmännische Berechnungen – Spiele – Operations Research – Statistik

Apple ProDOS für Aufsteiger

Band 1
von Ulrich Stiehl
1984, 202 S., kart., DM 28,–

ProDOS ist das neue „professionelle DOS" (Professional Disk Operating System) für den Apple IIe sowie den mit einer Language Card ausgestatteten Apple II Plus. Band 1 befaßt sich mit den theoretischen Grundlagen von ProDOS, der internen und externen Speicherorganisation und enthält grundlegende Beispielprogramme für Assembler-Programmierer sowie generelle Untersuchungen zum BASIC-SYSTEM. Da ProDOS über erheblich vielfältigere und leistungsfähigere, zugleich jedoch erheblich kompliziertere Dateistrukturen verfügt, sind theoretische Kenntnisse von ProDOS unabdingbar, wenn man die Features von ProDOS voll ausschöpfen will.
Aus dem Inhalt:
Ein erster Überblick – ProDOS und DOS 3.3 – Interne Speicherorganisation – Externe Speicherorganisation – MLI (Machine Language Interface) – ProDOS für Applesoft-Programmierer

Beachten Sie die Buch-Shop-Karte

Fourier-Analyse

von Peter Walber

Jeder, der einmal vor das Problem gestellt war, eine Funktion zu approximieren, wird sicher bereits auf die Fourier-Analyse gestoßen sein, vor allem bei der Darstellung von unstetigen periodischen Funktionen, also mathematisch schwierig darstellbaren Kurvenzügen.
Grundlage des Programms ist die von Fourier im Jahre 1822 aufgestellte Theorie, die besagt, daß sich beliebige periodische Funktionsverläufe als Linearkombination von Sinus- und Cosinus-Termen darstellen lassen. Mathematisch formuliert sieht die Theorie so aus:

$$f(t) = a_0 + \sum_{k=1}^{\infty} a_k \cos(k\omega t) + b_k \sin(k\omega t)$$

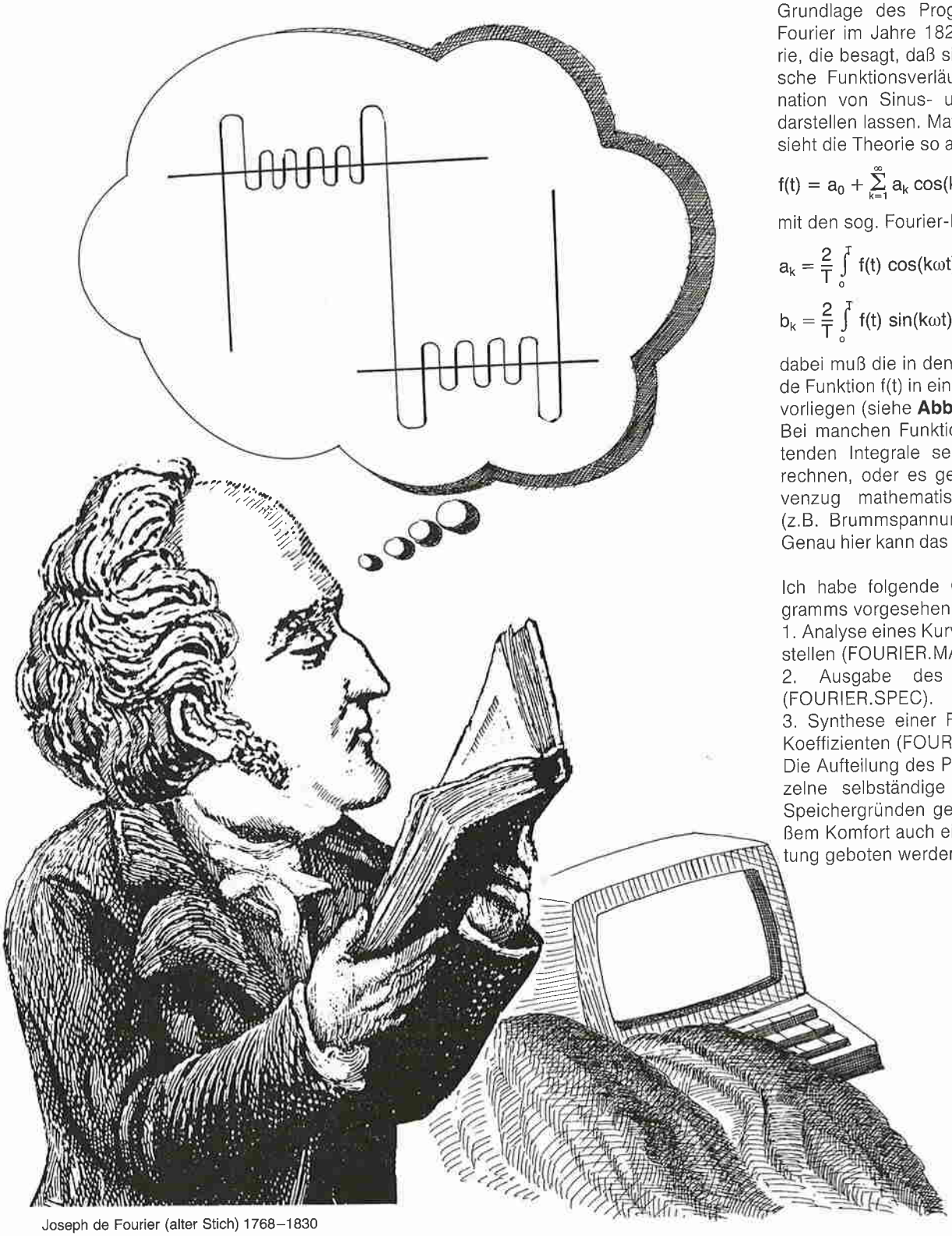
mit den sog. Fourier-Koeffizienten:

$$a_k = \frac{2}{T} \int_0^T f(t) \cos(k\omega t)\, dt$$

$$b_k = \frac{2}{T} \int_0^T f(t) \sin(k\omega t)\, dt$$

dabei muß die in den Integralen auftretende Funktion f(t) in einem Intervall $0 < t < T$ vorliegen (siehe **Abb. 1**).
Bei manchen Funktionen sind die auftretenden Integrale sehr schwierig zu berechnen, oder es gelingt nicht, den Kurvenzug mathematisch zu beschreiben (z.B. Brummspannung eines Netzteiles). Genau hier kann das Programm einsetzen.

Ich habe folgende Gliederung des Programms vorgesehen:
1. Analyse eines Kurvenzuges nach Stützstellen (FOURIER.MAIN).
2. Ausgabe des Frequenzspektrums (FOURIER.SPEC).
3. Synthese einer Funktion aus Fourier-Koeffizienten (FOURIER.SYN).
Die Aufteilung des Programms in drei einzelne selbständige Module wurde aus Speichergründen gewählt, da neben großem Komfort auch eine grafische Auswertung geboten werden sollte.



Joseph de Fourier (alter Stich) 1768–1830

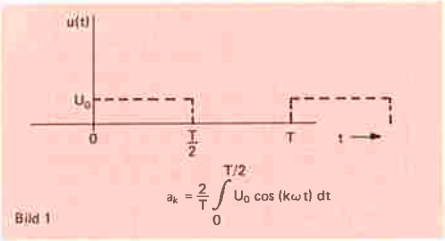


Bild 1

Das Modul FOURIER.MAIN

Das Modul gliedert sich in die folgenden Teile:
– Initialisierung und Hauptmenü (Zeilen 1000–1420).
– Eingabe der Stützstellen wahlweise von Diskette, Tastatur oder von einer analytischen Funktion (Zeilen 1450–1750).
– Berechnung der Fourier-Koeffizienten (Zeilen 1760–2230).
– Ausgabe der Ergebnisse und Plot der Funktion (Zeilen 2240–2730 und 2980–3500).
– Abspeichern der berechneten Daten (ab Zeile 3630).
Es folgt nun eine genauere Beschreibung der wichtigen Teile im Programm, vermischt mit den unbedingt notwendigen Hintergrundinformationen über die Fourier-Analyse.

Eingabe der Stützstellen

Da wir nicht in der Lage sind, eine unendliche Reihe zu berechnen, müssen wir eine Obergrenze festlegen. Wir brechen also die Reihe ab und approximieren nur mit einem trigonometrischen Polynom vom Grade n.

$$f(t) = a_0 + \sum_{k=1}^{n} a_k \cos(k\omega t) + b_k \sin(k\omega t)$$

Der Algorithmus von Runge (1), der hier verwendet wird, erwartet 2 · n Stützstellen. Diese müssen dem Graphen in äquidistanten Intervallen entnommen werden. Betrachten wir hierzu die in **Abb. 2** dargestellte Rechteckfunktion. Die Funktionswerte f(0) = 1, f(1) = 1, f(2) = 1, f(3) = 1, f(4) = 1 f(5) = 0, f(6) = 0, f(7) = 0 können dem Graphen entnommen werden. Hierbei ist dt der Abstand auf der t-Achse zwischen dem Wert f(i) und f(i + 1).

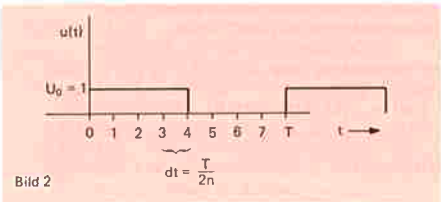


Bild 2

Berechnung der Koeffizienten

Der hier verwendete Algorithmus verlangt vom Programm die Aufsummierung nach den folgenden Formeln:

$$a_k = \frac{1}{n} \sum_{k=0}^{2n-1} f(t) \cos\left(\frac{k\pi t}{n}\right)$$

$$b_k = \frac{1}{n} \sum_{k=0}^{2n-1} f(t) \sin\left(\frac{k\pi t}{n}\right)$$

Der Beweis folgt aus den Orthogonalitätsrelationen für trigonometrische Funktionen und ist hier sicher nicht von Interesse (2). Dieses Gleichungssystem ist nur mit großem Rechenaufwand lösbar. Wir benutzen deshalb die Symmetrie-Eigenschaften der Sinus- bzw. Cosinus-Funktion.

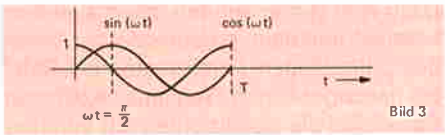


Bild 3

Wie aus **Abb. 3** ersichtlich ist, wiederholen sich alle Funktionswerte nach x = π/2. Weiter gilt sin(x) = cos(π/2 - x). Wir bilden daher zunächst die zweite Halbperiode auf die erste ab und danach die zweite Viertelperiode auf die erste. Dies wurde in den Zeilen 4500–4600 verwirklicht. Die zugehörigen Sinus-Werte werden in den Zeilen 1610–1640 im Feld X(I) abgelegt. In den Zeilen 1750–2220 erfolgt dann die eigentliche Berechnung der Fourier-Koeffizienten. Dieser Algorithmus ist von Sarnow in (4) beschrieben worden. Ab Zeile 2230 werden die Ergebnisse ausgegeben, entweder auf den Bildschirm oder sofort zum Drucker. Die Druckerausgabe muß eventuell an einen anderen Drucker angepaßt werden (Zeile 2800 und 4620).
Eine weitere Option dieses Moduls ist die Berechnung der Fourier-Sinus-Reihe. Dies ist die Darstellung der Reihe in der Form:

$$f(t) = c_0 + \sum_{k=1}^{\infty} c_k \sin(k\omega t + \varphi_k)$$

Grafik-Auswertung

Damit das Programm nicht nur zur Schaffung von Datengräbern dient, wurde eine komfortable Grafik-Auswertung eingebaut (2400 ff.). Dabei wird zuerst die Funktion mit wählbarem Grad des trigonometrischen Polynoms geplottet. Dadurch kann man sich die Genauigkeit der Approximation Schritt für Schritt ansehen. Danach wird die Funktion noch einmal nach den eingegebenen Stützstellen gezeichnet. Die Beschriftung der Achsen erfolgt durch POKEs von bestimmten Bitmustern direkt in den Grafik-Speicher.

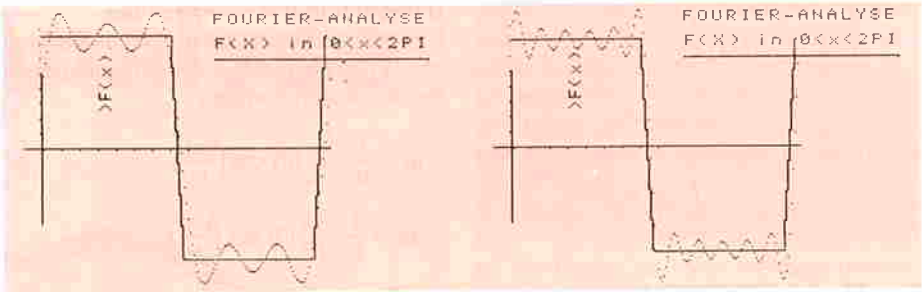
Speicherung der Ergebnisse

Im Anschluß daran besteht die Möglichkeit, den Plot als Bild-File abzuspeichern oder eine Hardcopy auszugeben. Weiter wird erfragt, ob die Koeffizienten oder die Funktionswerte als Datei für spätere Betrachtungen gesichert werden sollen. Man kann sich damit eine regelrechte Bibliothek von Funktionen anlegen. Ab Zeile 3630 folgen dann einige Unterprogramme, die die beschriebenen Möglichkeiten verwirklichen.
Eine Besonderheit liegt in der Handhabung der Daten-Files. Dem Dateinamen wird automatisch ein Suffix angehängt. Sollen also Koeffizienten abgespeichert werden, so gibt man z.B. „DREIECK" als Dateiname an; auf der Diskette erscheint dann eine Datei mit dem Namen „DREIECKKOEFF". Will man die Daten wieder einlesen, dann muß man auf die Frage nach der Datei den vollen Namen angeben, also „DREIECKKOEFF".
Im Programm wurde als maximaler Grad der Fourier-Reihe 24 festgelegt. Dies ist sicher für den normalen Fall ausreichend. Ein Nachteil liegt ohne Zweifel in der Ungenauigkeit bei der Analyse sehr unstetiger Funktionen. Das Verfahren wird mit zunehmender Stützstellenzahl natürlich exakter. Genauere Untersuchungen zeigen jedoch, daß der Rechenaufwand nicht proportional steigt, sondern viel stärker. Außerdem kommt noch die mangelhafte Rechengenauigkeit des Basic-Interpreters zum Tragen. Abhilfe schafft zumindest teilweise der Übergang auf ein FORTRAN-Programm mit Double-Precision (3). Die hier erzielten Ergebnisse stellen aber sicher einen sinnvollen Kompromiß dar.
Das Programm wurde für 80 Zeichen/Zeile geschrieben und benutzt deshalb Ctrl-L statt des HOME-Befehls. Da beim IIe mit erweiterter 80-Zeichen-Karte und dem IIc die HGR2 in Verbindung mit der 80-Zeichendarstellung nicht aufgerufen werden kann, wird vor der Grafik auf Normalausgabe zurückgeschaltet.
Durch den menügesteuerten Aufbau sind Bedienungsfehler weitgehend ausgeschlossen.
Beispiele 1 zeigt die Möglichkeit des Programms. Bei der dargestellten Rechteckfunktion wurde der Grad n = 5 und n = 12 gewählt. Hierbei wird die unterschiedliche Güte der Approximation deutlich.

Die folgenden Module FOURIER.SYN und FOURIER.SPEC können aus Platzgründen nicht abgedruckt werden und sind daher nur auf der Peeker Sammeldiskette enthalten.

Das Modul FOURIER.SYN

Der Aufruf erfolgt im Hauptmenü mit „2" für Synthese. Dieser Teil bietet die Möglichkeit,aus früher berechneten Koeffizienten eine Funktion darzustellen. Dies geschieht mit wählbarem Grad des trigonometrischen Polynoms. Das Modul selbst besteht zum größten Teil aus Teilen des Hauptprogramms FOURIER.MAIN.



Beispiel 1: n = 5 (links) und n = 12 (rechts)

Das Modul FOURIER.SPEC

Der Aufruf erfolgt mit der Wahl von „3" für Frequenzspektrum im Hauptmenü. Bevor das Modul genauer beschrieben wird, wollen wir zunächst die nötige Theorie kurz erläutern. Das „Spektrum" einer Funktion gibt Auskunft über die Beträge der enthaltenen Schwingungsanteile. Man kann also aus der Darstellung sofort entnehmen, welche Frequenzanteile in welcher Stärke vertreten sind. Die Verwirklichung sieht nun wie folgt aus:

– Eingabe der Koeffizienten von wahlweise Disk oder Tastatur (Zeilen 1000–1160).

– Vorbereiten des Plots (Zeilen 1160–1420). Hierzu gehört die Erstellung des Koordinatenkreuzes und eines Binär-Feldes zur Beschriftung des Plots.

– Zeichnen des Plots (Zeilen 1450–1730). Dazu wird vorher auf den maximalen Wert normiert und dann proportional zur Größe des betreffenden Koeffizienten ein Strich bei der zugehörigen Frequenz gezeichnet. Dabei sind die Koeffizienten a(I) durchgezogen und die b(I) gestrichelt dargestellt.

– Abspeichern des Bildes, der Koeffizienten und die Hardcopy-Möglichkeit (Zeilen 1750–Ende).

Die Zeilen, die die Files anlegen, sind bereits in FOURIER.MAIN vorhanden.

Benötigte System-Konfiguration: Apple II, DOS 3.3, 80-Zeichenkarte, Drucker mit Grafik-Interface.

Literatur:

(1) Zurmühl, Praktische Mathematik für Ingenieure und Physiker, Springer 1965

(2) Hamming, Numerical Methods for Scientists and Engineers, McGraw-Hill, 1962

(3) McCracken, FORTRAN

(4) Sarnow, mc 1/84, S. 82–85

FOURIER.MAIN

```
1000 REM ***    Fourier-Analyse    ***
1010 REM *** Peter Walber, Mai '84 ***
1020 REM
1030 HIMEM: 16384
1040 D$ = CHR$ (4)
1050 PRINT D$;"PR#3"
1060 DIM Y(50),X(50),YY(50),SS(25),DS(25),DD(25)
1070 DIM A(25),B(24),Y1(160),ZZ(50),SD(25)
1075 DIM AA(25),PHI(25)
1080 PRINT CHR$ (12)
1090 FOR I = 0 TO 79: PRINT "-";: NEXT I
1100 PRINT "                          Fourier-Analyse"
1110 FOR I = 0 TO 79: PRINT "-";: NEXT I: PRINT
1120 PRINT "  ";
1130 PRINT "Optionen:": PRINT "  ";: FOR I = 0 TO 72: PRINT
     "-";: NEXT I: PRINT
1140 PRINT "  ";
1150 PRINT TAB(20)"Analyse einer einzugebenden Funktion
     1";
1155 PRINT
1160 PRINT "  ";
1180 FOR I = 1 TO 71: PRINT " ";: NEXT I
1185 PRINT
1200 PRINT "  ";
1210 PRINT TAB(20)"Synthese einer gesuchten Funktion
     2";: PRINT
1220 PRINT "  ";
1230 FOR I = 1 TO 71: PRINT " ";: NEXT I
1240 PRINT
1250 PRINT "  ";
1260 PRINT TAB(20)"Frequenzspektrum ausgeben
     3";: PRINT
1270 PRINT "  ";
1280 FOR I = 1 TO 71: PRINT " ";: NEXT I
1290 PRINT
1300 PRINT "  ";
1310 PRINT TAB(20)"Ende
     4";: PRINT
1320 PRINT "  ";
1330 FOR I = 1 TO 71: PRINT " ";: NEXT I
1340 PRINT
1350 FOR I = 0 TO 79: PRINT "-";: NEXT I
1360 PRINT TAB(20)"Bitte die Eingabe der Programmwahl"
1380 FOR I = 0 TO 79: PRINT "-";: NEXT I: PRINT
1390 INPUT "1,2,3,4 ";IN
1400 IF IN < 1 OR IN > 4 THEN 1390
```

```
1410 ON IN GOSUB 1460,3520,3570,3610
1420 GOTO 1100
1430 REM
1440 REM ***     Start der Analyse     ***
1450 REM *** Eingabe der Stützstellen ***
1460 REM
1470 PRINT CHR$ (12)
1480 PRINT "Analyse einer Funktion nach Runge"
1490 PRINT : PRINT "Notwendige Bedingung : Grad der Reihe
     geradzahlig "
1500 PRINT
1510 PRINT : PRINT "Eingabe der Stützstellen von (D)isk,
     (T)astatur oder von einer": PRINT : PRINT
     "analytischen (F)unktion"
1520 GET IN$
1530 IF IN$ = "D" THEN GOSUB 3940:DT$ = "J": GOTO 1590
1540 IF IN$ < > "T" AND IN$ < > "F" THEN 1460
1550 PRINT
1560 INPUT "Grad der Fourier-Reihe: ";N
1570 IF N / 2 < > INT (N / 2) THEN PRINT : PRINT "Bitte
     „N„ muß durch zwei teilbar sein.": PRINT : GOTO 1560
1580 IF N > 24 THEN PRINT : PRINT "Nmax=24":N = 24
1590 P = N / 2:M = 2 * N
1600 IF IN$ = "F" THEN GOSUB 4310
1610 DX = 3.141592654 / N
1620 FOR I = 0 TO P
1630 X(I) = SIN (I * DX):X(N - I) = X(I):X(N + I) = -
     X(I):X(M - I) = X(N + I)
1640 NEXT I
1650 IF DT$ = "J" THEN 1700
1660 IF IN$ = "F" THEN 1700
1670 FOR I = 0 TO M - 1
1680 PRINT "Y(";I;")= ";: INPUT "";Y(I)
1690 NEXT I
1700 FOR I = 0 TO M - 1: PRINT "Y(";I;")= ";Y(I): NEXT I:
     PRINT
1710 PRINT : INPUT "Änderung eines eingegebenen Wertes?
     (J/N)";IN$: IF IN$ < > "J" THEN 1740
1720 PRINT : INPUT "Index des zu ändernden Wertes";I
1730 PRINT : INPUT "Neuer Wert";Y(I): GOTO 1710
1740 REM
1750 REM *** Start der Berechnung ***
1760 REM
1770 Y(0) = Y(0) / 2:Y(M) = Y(0):Y(N) = Y(N) / 2
1780 GOSUB 4530
1790 A1 = 0:A2 = 0
1800 FOR I = 0 TO P STEP 2
1810 A1 = A1 + SS(I)
```

```
1820 NEXT I
1830 FOR I = 1 TO P STEP 2
1840 A2 = A2 + SS(I)
1850 NEXT I
1860 A(0) = (A1 + A2) / M:A(N) = (A1 - A2) / M
1870 FOR N1 = 1 TO P STEP 2
1880 A1 = 0:A2 = 0:B1 = 0:B2 = 0
1890 L = 0:K = P
1900 FOR I = 0 TO P STEP 2
1910 A1 = A1 + DS(I) * X(K):B2 = B2 + SD(I) * X(L)
1920 L = L + 2 * N1: IF L = M THEN L = L - M
1930 K = K + 2 * N1: IF K = M THEN K = K - M
1940 NEXT I
1950 PRINT ">";
1960 L = N1:K = P + N1
1970 FOR I = 1 TO P STEP 2
1980 A2 = A2 + DS(I) * X(K):B1 = B1 + SD(I) * X(L)
1990 L = L + 2 * N1: IF L = M THEN L = L - M
2000 K = K + 2 * N1: IF K = M THEN K = K - M
2010 NEXT I
2020 A(N1) = (A1 + A2) / N:A(N - N1) = (A1 - A2) / N
2030 B(N1) = (B1 + B2) / N:B(N - N1) = (B1 - B2) / N
2040 NEXT N1
2050 FOR N1 = 2 TO P STEP 2
2060 A1 = 0:A2 = 0:B1 = 0:B2 = 0
2070 L = 0:K = P
2080 FOR I = 0 TO P STEP 2
2090 A1 = A1 + SS(I) * X(K):B2 = B2 + DD(I) * X(L)
2100 L = L + 2 * N1: IF L = M THEN L = L - M
2110 K = K + 2 * N1: IF K = M THEN K = K - M
2120 NEXT I
2130 L = N1:K = P + N1
2140 FOR I = 1 TO P STEP 2
2150 A2 = A2 + SS(I) * X(K):B1 = B1 + DD(I) * X(L)
2160 L = L + 2 * N1: IF L = M THEN L = L - M
2170 K = K + 2 * N1: IF K = M THEN K = K - M
2180 NEXT I
2190 A(N1) = (A1 + A2) / N:A(N - N1) = (A1 - A2) / N
2200 B(N1) = (B1 + B2) / N:B(N - N1) = (B1 - B2) / N
2210 NEXT N1
2220 REM
2230 REM *** Ende der Berechnung ***
2240 REM *** Beginn der Ausgabe  ***
2250 REM
2260 INPUT "Ergebnis auf den (D)rucker oder (B)ildschirm?
     ";IN$
2270 IF IN$ = "D" THEN GOSUB 3300: GOTO 2351
2280 IF IN$ < > "B" THEN 2260
2290 PRINT "Ergebnis:": PRINT "--------": PRINT
2300 PRINT : PRINT "a(0)=";A(0): PRINT
2310 FOR I = 1 TO N
2320 PRINT "A(";I;")=";A(I), TAB(40)"B(";I;")=";B(I)
2330 NEXT I: PRINT : INPUT "Weiter --->> CR <<---";IN$
2340 PRINT : INPUT "Ausgabe auf den Drucker? (J/N)";DD$
2350 IF DD$ = "J" THEN GOSUB 3300: PRINT
2351 PRINT "Darstellung als Fourier-Sinus-Reihe:": PRINT
     "-----------------------------------"
2352 FOR I = 1 TO N: LET AA(I) = SQR (A(I) ↑ 2 + B(I) ↑ 2)
2353 IF B(I) = 0 AND A(I) = 0 THEN PHI(I) = 0: GOTO 2365
2354 IF B(I) = 0 AND A(I) > 0 THEN PHI(I) = 0: GOTO 2365
2355 IF B(I) = 0 AND A(I) < 0 THEN PHI(I) = 180: GOTO 2365
2356 IF A(I) = 0 AND B(I) > 0 THEN PHI(I) = 90: GOTO 2365
2357 IF A(I) = 0 AND B(I) < 0 THEN PHI(I) = - 90: GOTO 2365
2360 LET PHI(I) = ATN (A(I) / B(I))
2361 LET PHI(I) = 180 * PHI(I) / 3.141592654
2365 NEXT I
2366 PRINT "C(0)=";A(0): PRINT
2367 FOR I = 1 TO N: PRINT "C(";I;")= ";AA(I);: HTAB 20:
     PRINT "Phi(";I;")= ";PHI(I): NEXT I
2368 PRINT : INPUT "Weiter -->>CR<<--";IN$
2369 REM
2370 REM *** Vorbereitung des Plots ***
2380 REM
2390 PRINT CHR$ (1)
2400 OM = 2 * 3.141592654: PRINT CHR$ (12)
2410 PRINT TAB(20)"Vorbereitung des Plots": PRINT
     TAB(20)"----------------------"
2420 PRINT : PRINT "Plotten der Funktion mit wählbarer
     Genauigkeit möglich": PRINT
2430 PRINT : INPUT "Annäherung einschließlich der Oberwelle
     Nr.:";G%: IF G% > N THEN G% = N
2440 IF G% > N THEN G% = N
2450 IF G% < 0 THEN PRINT : PRINT "Fehlerhafte Eingabe ":
     GOTO 2430
2460 PRINT : PRINT "Moment bitte": PRINT
2470 REM
2480 REM *** Approximation mit den oben berechneten
     Fourier-Koeffizienten ***
2490 REM
2500 FOR I1 = 0 TO 159
2510 S = A(0)
2520 FOR I2 = 1 TO G%
2530 S = S + A(I2) * COS (I2 * OM * I1 / 160) + B(I2) * SIN
     (I2 * OM * I1 / 160)
2540 NEXT I2
2550 Y1(I1) = S
2560 NEXT I1
2570 J = 1
2580 PRINT ">";
2590 GOSUB 3000
2600 FOR I = 0 TO 159
2610 HPLOT I + 10,96 - 90 * Y1(I) / Y0
2620 NEXT I
2630 INPUT "---->> CR ";IN$
2640 FOR I = 0 TO M - 1
2650 IF ABS (Y(I)) > Y0 THEN Y0 = ABS (Y(I))
2660 NEXT I
2670 FOR I = 0 TO M - 1
2680 C = 0:D = 0:E = 0:F = 0
2690 C = 10 + 2 * I * 79 / M:D = 96 - 90 * Y(I) / Y0:E = 10
     + 2 * (I + 1) * 79 / M:F = 96 - 90 * Y(I + 1) / Y0
2700 HPLOT C,D TO E,F
2710 NEXT I
2720 GET ZZ$: PRINT CHR$ (12)
2730 PRINT D$;"PR#3": TEXT
2740 INPUT "Abspeichern des Plots als Bild-File? (J/N)";BF$
2750 IF BF$ < > "J" THEN 2790
2760 PRINT
2770 INPUT "Auf Slot=";S: PRINT : INPUT "und Drive=";D:
     PRINT
2780 PRINT D$;"BSAVE FPICl,A$4000,L$1FFF,S";S;",D";D
2790 PRINT : INPUT "Hardcopy des Plots erstellen?
     (J/N)";HK$
2800 IF HK$ = "J" THEN GOSUB 4620
2810 PRINT
2820 PRINT : INPUT "Erstellen eines weiteren Plots mit
     anderer Genauigkeit? (J/N)";NP$
2830 IF NP$ = "J" THEN PRINT CHR$ (12): GOTO 2410
2840 PRINT
2850 INPUT "Fourier-Koeffizienten abspeichern? (J/N)";IN$
2860 IF IN$ = "N" THEN PRINT : GOTO 2900
2870 IF IN$ < > "J" THEN 2850
2880 GOSUB 3630
2890 PRINT
2900 INPUT "Eingegebene Funktionswerte speichern?
     (J/N)";IN$
2910 PRINT FRE (0)
2920 IF IN$ = "N" THEN 1080
2930 IF IN$ < > "J" THEN 2900
2940 FOR I = 0 TO M - 1:Y1(I) = Y(I): NEXT I
2950 NF = N: GOSUB 3790
2960 GOTO 1080
2970 REM
2980 REM *** Bildrahmen generieren ***
2990 REM
3000 X0 = 2 * 3.141592654:XU = 0:Y0 = - 1.E25:YU = 1.E25
3010 FOR I = 0 TO 159
3020 IF Y1(I) > Y0 THEN Y0 = Y1(I)
3030 IF Y1(I) < YU THEN YU = Y1(I)
3040 NEXT I
3050 FOR I = 0 TO 20: PRINT : NEXT I
3060 PRINT : PRINT TAB(10)"Ymax= ";Y0
3070 PRINT : PRINT TAB(10)"Ymin= ";YU
3080 PRINT
3090 IF ABS (YU) > Y0 THEN Y0 = ABS (YU)
3100 PRINT TAB(10)"XU= ";XU
3110 PRINT : PRINT TAB(10)"X0= ";X0
3120 PRINT : PRINT "Nährung der Funktion nach Fourier
     einschließlich Oberwelle Nr.: ";G%: PRINT
3130 PRINT : PRINT "Weiter --->> CR <<---";: GET XX$: PRINT
     FRE (0)
3140 PRINT D$;"PR#0": HGR2 : HCOLOR= 7
3150 GOSUB 5000: REM *** Bildrahmen generieren ***
3160 FOR I = 0 TO 100
3170 HPLOT 10,46 + I
3180 NEXT I
3190 FOR I = 0 TO 170
3200 HPLOT I,96
3210 NEXT I
3220 FOR I = 0 TO 80 STEP 10
3230 HPLOT 9,96 - I
3240 NEXT I
3250 FOR I = 11 TO 91 STEP 10
3260 HPLOT I,97
3270 NEXT I
3280 RETURN
```

```
3290 REM
3300 REM ********* Druckerausgabe ********
3301 REM Die hier verwendeten Druckersteuerzeichen gelten
     für den STAR
3302 REM GEMINI-Drucker und sind an den verwendeten Drucker
     anzupassen.
3303 REM CHR$(9);"G2" gibt eine Hardcopy der HGR2-Seite auf
     den Drucker.
3304 REM CHR$(87);CHR$(0) schaltet auf Normalbreite zurück.
3305 REM CHR$(87);CHR$(1) schaltet auf Breitschrift um.
3310 REM
3320 PRINT D$;"PR#1": PRINT CHR$ (27); CHR$ (87); CHR$ (1)
3330 PRINT "F O U R I E R - A N A L Y S E "
3335 PRINT "-----------------------------"
3370 PRINT CHR$ (27); CHR$ (87); CHR$ (0)
3380 PRINT "Algorithmus: Runge": PRINT
3390 PRINT CHR$ (15)
3400 PRINT "Anzahl der eingegebenen Stützstellen: ";M
3410 PRINT : PRINT "Grad des trigonometrischen Polynoms: ";N
3415 PRINT "Koeffizienten:"
3420 PRINT "a(0)= ";A(0)
3430 PRINT : FOR I = 1 TO N
3440 PRINT TAB(10)"a(";I;")= ";A(I); CHR$ (27); CHR$ (98);
     CHR$ (40);"b(";I;")= ";B(I)
3450 PRINT
3460 NEXT I
3470 PRINT : PRINT "Fourier-Analyse": PRINT : PRINT "(c)
     Peter Walber, Mai '84"
3480 PRINT CHR$ (27); CHR$ (50)
3490 PRINT D$;"PR#3"
3491 REM CHR$(98);CHR$(40) setzt den nächsten Tabulator auf
     die 40. Spalte
3492 REM CHR$(50) Zeilenvorschub auf 1/6 Inch
3500 RETURN
3510 REM
3520 REM ******* Synthese ********
3530 REM
3540 PRINT : PRINT "FOURIER.SYN wird geladen"
3550 PRINT D$;"RUN FOURIER.SYN"
3560 REM
3570 REM ******** Spektrum *******
3580 REM
3590 PRINT : PRINT "FOURIER.SPEC wird geladen"
3600 PRINT D$;"RUN FOURIER.SPEC"
3610 END
3620 REM
3630 REM ******************************************
3640 REM
3650 PRINT CHR$ (12)
3660 PRINT "Abspeichern der Fourier-Koeffizienten"
3670 PRINT : INPUT "Filename:";DN$
3680 PRINT : INPUT "Slot:";S
3690 PRINT : INPUT "Drive:";D
3700 PRINT D$;"OPEN";DN$ + "KOEFF";",S";S;",D";D
3710 PRINT D$;"DELETE";DN$ + "KOEFF"
3720 PRINT D$;"OPEN";DN$ + "KOEFF"
3730 PRINT D$;"WRITE";DN$ + "KOEFF"
3740 PRINT N: PRINT A(0)
3750 FOR I = 1 TO N + 3: PRINT A(I): PRINT B(I): NEXT I
3760 PRINT D$;"CLOSE"
3770 RETURN
3780 REM
3790 REM ******************************************
3800 REM
3810 PRINT CHR$ (12)
3820 PRINT "Abspeichern der Funktionswerte"
3830 PRINT : INPUT "Filename:";DN$
3840 PRINT : INPUT "Slot:";S
3850 PRINT : INPUT "Drive:";D
3860 PRINT D$;"OPEN";DN$ + "WERTE";",S";S;",D";D
3870 PRINT D$;"DELETE";DN$ + "WERTE"
3880 PRINT D$;"OPEN";DN$ + "WERTE"
3890 PRINT D$;"WRITE";DN$ + "WERTE": PRINT NF: FOR I = 0 TO
     2 * NF + 2
3900 PRINT Y1(I): NEXT I
3910 PRINT D$;"CLOSE"
3920 RETURN
3930 REM
3940 REM ******************************************
3950 REM
3960 GOSUB 4230
3970 PRINT CHR$ (12)
3980 PRINT "Einlesen der Funktionswerte"
3990 PRINT : INPUT "Filename: ";DN$
4000 PRINT : INPUT "Slot:";S: PRINT : INPUT "Drive:";D
4010 PRINT D$;"OPEN";DN$;",S";S;",D";D
4020 PRINT D$;"READ ";DN$
4030 INPUT YZ$: PRINT :N = VAL (YZ$):P = N / 2:M = 2 * N
4040 PRINT
4050 FOR I = 0 TO M - 1: INPUT YZ$: PRINT :Y(I) = VAL
     (YZ$): NEXT I
4060 PRINT D$;"CLOSE"
4070 RETURN
4080 REM
4090 REM ******************************************
4100 REM
4110 D$ = CHR$ (4): GOSUB 4230
4120 PRINT CHR$ (12)
4130 PRINT "Einlesen der Fourier-Koeffizienten"
4140 PRINT : INPUT "Filename:";DN$
4150 PRINT : INPUT "Slot:";S
4160 PRINT : INPUT "Drive";D
4170 PRINT D$;"OPEN";DN$;",S";S;",D";D
4180 PRINT D$;"READ";DN$
4190 INPUT N$: PRINT :N = VAL (N$): INPUT AB$: PRINT :A(0)
     = VAL (AB$)
4200 FOR I = 1 TO N: INPUT YZ$: PRINT :A(I) = VAL (YZ$):
     INPUT YZ$: PRINT :B(I) = VAL (YZ$): NEXT I
4210 PRINT D$;"CLOSE"
4220 RETURN
4230 PRIN
4400 STOP
4410 DEF FN F(I) = 10 * EXP ( - I ↑ 2)
4420 FOR I = 0 TO 10: PRINT : NEXT I
4430 PRINT "Da die Fourier-Analyse nur periodische
     Funktionen auswertet, muß dem": PRINT : PRINT
     "Programm mitgeteilt werden, wo die Periodendauer T
     liegen soll."
4440 INPUT "Obere Grenze=";TMAX
4450 LET DT = TMAX / M:DT = INT (DT)
4460 FOR I = 0 TO TMAX - DT
4470 Y(I) = FN F(I)
4480 NEXT I
4490 RETURN
4500 REM
4510 REM *** Abbildung auf den 1. Quadranten
4520 REM
4530 FOR I = 0 TO N
4540 ZZ(I) = Y(I) + Y(M - I):YY(I) = Y(I) - Y(M - I): NEXT I
4550 ZZ(P) = ZZ(P) / 2:YY(P) = YY(P) / 2
4560 Y(0) = 2 * Y(0):Y(M) = Y(0):Y(N) = 2 * Y(N)
4570 FOR I = 0 TO N / 2
4580 SS(I) = ZZ(I) + ZZ(N - I):DS(I) = ZZ(I) - ZZ(N - I)
4590 SD(I) = YY(I) + YY(N - I):DD(I) = YY(I) - YY(N - I):
     NEXT I
4600 RETURN
4610 REM
4620 REM *** Hardcopy ***
4630 REM
4640 PRINT : INPUT "Plot vergrößert zur Ausgabe? (J/N)";VG$
4650 IF VG$ = "J" THEN 4670
4660 PRINT D$;"PR#1": PRINT CHR$ (9);"G2": PRINT D$;"PR#3":
     RETURN
4670 PRINT D$;"PR#1": PRINT CHR$ (9);"G2RD": PRINT
     D$;"PR#3": RETURN
5000 REM *** Bildrahmen ***
5010 BASADR = 16407
5020 FOR J = 1 TO 8
5030 FOR I7 = 0 TO 7
5040 READ AL%
5050 POKE BASADR + 1024 * I7,AL%
5060 NEXT I7
5070 BASADR = BASADR + 1
5080 NEXT J
5081 BASADR = 16836
5082 FOR I7 = 0 TO 7
5083 READ AL%
5084 POKE BASADR + 1024 * I7,AL%
5085 NEXT I7
5090 RESTORE
5100 BASADR = 16384
5110 FOR I7 = 0 TO 7
5120 READ AL%
5130 POKE BASADR + 1024 * I7,AL%
5140 NEXT I7
5150 RESTORE
5160 HPLOT 160,9 TO 224,9
5170 RETURN
5180 DATA 0,62,2,2,30,2,2,2: REM F
5190 DATA 0,34,34,34,34,34,34,28: REM U
5200 DATA 0,34,34,38,42,50,34,34: REM N
5210 DATA 0,34,18,10,6,10,18,34: REM K
5220 DATA 0,62,8,8,8,8,8,8: REM T
5230 DATA 0,28,8,8,8,8,8,28: REM I
5240 DATA 0,28,34,34,34,34,34,28: REM O
5250 DATA 0,34,34,38,42,50,34,34: REM N
5260 DATA 0,0,0,34,20,8,20,34: REM x
```

Applesoft-Erweiterungen

von Markus Enzinger

Eine Sammlung von Ampersand-Routinen

Mit Hilfe des Assembler-Programms AS.ERWEITERUNG werden insgesamt 18 neue Befehle in Applesoft implementiert, die von jedem Basic-Programm aus aufgerufen werden können.
Nach dem Eingeben des Assembler-Programms speichern Sie dieses bitte unter dem Namen **AS.ERWEITERUNG** mit „BSAVE AS.ERWEITERUNG, A$9378, L$288" auf Diskette ab. Das Programm kann dann von Applesoft aus mit einem BRUN jederzeit wieder eingeladen werden und liegt nach dem Einladen unter den DOS-Puffern, d.h. von Adresse $95FF abwärts. Es schützt sich nach dem Starten (durch BRUN oder CALL 38399) selbst vor Überschreiben durch Stringvariablen und initialisiert dann den &-Vektor, der von Applesoft aus aufgerufen wird, sobald der Interpreter auf das „&"-Zeichen trifft. Dieser Vektor steht in den Speicherzellen $03F5 bis $03F7. In diesen Bereich wird ein Sprungbefehl zu dem Programm geschrieben, welches die Befehle nach dem „&" verarbeitet.

Die Erläuterung der einzelnen Befehle

& GOTO stellt dem Benutzer die Möglichkeit zur Verfügung, ein berechnetes GOTO durchzuführen. Als Zeilennummer darf hierfür jeder beliebige numerische Ausdruck eingesetzt werden. Beispiele hierzu:
& GOTO 10 * A + 25
& GOTO A(X) * 10 + 50.

& GOSUB bildet einen berechneten Sprung in ein Unterprogramm. Der Geltungsbereich für den numerischen Ausdruck ist identisch mit dem des Befehls „& GOTO".

& RESTORE hat im Prinzip die gleiche Wirkung wie der Applesoft-Befehl „RESTORE" mit dem Unterschied, daß der DATA-Zeiger, der auf die momentan zu lesende Information gerichtet ist, nicht auf das erste vorhandene DATA-Statement zurückgesetzt wird, sondern auf jede beliebige DATA-Zeile gerichtet werden kann. Der numerische Ausdruck für die Zeilennummer muß den gleichen Bedingungen entsprechen wie der von „& GOTO".

& HGR bewirkt, daß die erste Seite für hoch aufgelöste Grafik sichtbar wird, ohne diese dabei zu löschen. Genauso wie bei dem „HGR"-Befehl in Applesoft werden auch hier die untersten vier Zeilen für Textausgabe reserviert. Dieser Befehl erspart das ansonsten notwendige „Herumpoken" in den Adressen für die Bildschirmausgabe. „& HGR" entspricht den folgenden Basic-Zeilen:
POKE -16304,0
POKE -16301,0
POKE -16300,0
POKE -16297,0
POKE 230,32.
Außerdem wird in der Zero-Page ein Flag gesetzt, welches bewirkt, daß alle nachfolgenden Grafikkommandos wie HPLOT usw. auf der ersten Grafikseite ausgeführt werden.

& HGR2 bewirkt dasselbe wie „& HGR", bezogen auf die zweite Seite für hoch aufgelöste Grafik. Allerdings wird hier, wie auch beim Applesoft-Befehl „HGR2", kein Platz für Textausgaben reserviert.

& GR schaltet die Bildschirmausgabe auf die erste Seite für niedrig aufgelöste Grafik um, ähnlich wie der Befehl „GR" in Applesoft, jedoch ebenfalls ohne diese zu löschen. Die unteren vier Zeilen sind für Textausgaben reserviert.

& INVERSE bewirkt ein Invertieren des Textbildschirmes. Invertieren bedeutet, daß die Hinter- und Vordergrundfarbe jedes einzelnen Zeichens gegeneinander vertauscht werden. Ein zweimaliges Anwenden dieses Befehls stellt wieder den Ursprungszustand des Textbildschirmes her. (Dieser Befehl sollte bei der Benutzung eines Peripheriegeräts nicht angesprochen werden, da der gesamte Bereich von $0400 bis $07FF invertiert wird und somit auch die von den Slots benutzten „Screenholes" ihren Wert ändern. Dies führt z.B. zu einer Neukalibrierung des Schreib/Lesekopfs bei einem angeschlossenen Diskettenlaufwerk. Anm. der Red.)

& INVERSE-H invertiert die zuletzt angesprochene Seite für hoch aufgelöste Grafik. Man sollte darauf achten, daß vor der Verwendung dieses Befehls schon eine der beiden Grafikseiten angesprochen wurde (durch „HGR", „HGR2", „& HGR" oder „& HGR2"). Der Grund dafür liegt in der Tatsache, daß in der Zero-Page ein Flag existiert, welches dem Interpreter sagt, auf welcher Grafikseite momentan gearbeitet wird. Ist noch keine Grafikseite angesprochen worden, so enthält dieses Flag einen Zufallswert und es könnte passieren, daß irgendwo im RAM Speicherzellen invertiert und dadurch Daten zerstört werden oder gar das System abstürzt.

& FLASH bewirkt ein Aufblitzen des Text- bzw. Lores-Grafik-Bildschirmes. Dieser Befehl ist besonders bei Spielen gut anwendbar, wenn sog. „Crashs" dargestellt werden sollen. Programmtechnisch wurde dieser Befehl realisiert, indem einfach die Invertierungsroutine zweimal hintereinander aufgerufen wird.

& SOUND h,d gibt über den internen Lautsprecher des Apple II den Ton h mit der Dauer d aus. Der Parameter h stellt also die Tonhöhe dar, d die Tondauer. H = 1 ergibt den tiefstmöglichen, h = 255 den höchstmöglichen Ton. D = 1 bewirkt einen sehr kurzen, d = 255 dagegen einen langen Ton. Für h und d sind sowohl Zahlen als auch Variablen einsetzbar. Beispiele hierfür:
SOUND 25,200
SOUND NH,ND + 100.

& SCROLL-R schiebt den gesamten Bildschirminhalt der Textseite um eine Spalte nach rechts. Mit Hilfe des SCROLL-Befehls ließe sich z.B. ein Programm zur Erzeugung von Laufschriften o.ä. recht einfach erstellen.

& SCROLL-L schiebt den Bildschirminhalt um eine Spalte nach links.

& SCROLL-U schiebt den Bildschirminhalt um eine Zeile nach oben.

& SCROLL-D schiebt den Bildschirminhalt um eine Zeile nach unten.

& SCROLL-GR schiebt den Bildschirminhalt der niedrig aufgelösten Grafik um eine Spalte nach rechts.

& SCROLL-GL schiebt den Bildschirminhalt der niedrig aufgelösten Grafik um eine Spalte nach links.

& SCROLL-GU schiebt den Bildschirminhalt der niedrig aufgelösten Grafik um eine Zeile nach oben.

& SCROLL-GD schiebt den Bildschirminhalt der niedrig aufgelösten Grafik um eine Zeile nach unten.

Die Grafik-SCROLL-Befehle eignen sich in erster Linie zur Erstellung von Spielprogrammen in Basic, wie z.B. Scramble oder Autorennen, die eine schnelle Bewegung der Hintergrundlandschaft verlangen. Die Text-SCROLL-Befehle sowie der INVERSE- und FLASH-Befehl arbeiten nur bei der 40 Z/Z-Ausgabe.

AS.ERWEITERUNG

```
                  1   ************************************
                  2   *                                  *
                  3   *     18 neue Applesoft-Befehle    *
                  4   *                                  *
                  5   *              von                 *
                  6   *         MARKUS ENZINGER          *
                  8   *                                  *
                  9   ************************************
                  10  *
                  11          ORG  $9378       ; Programm-Startadresse
                  12  *
                  13  * -------------- Label-Definition ----------------
                  14  *
                  15  * Zuordnung der Tokens, d.h. der Zahlen, die der Inter-
                  16  * preter für die jeweiligen Befehle verwendet
                  17  *
                  18  FLASHØ   EQU  159        ; Token von FLASH
                  19  INVERSEØ EQU  158        ; Token von INVERSE
                  20  GOSUBØ   EQU  176        ; Token von GOSUB
                  21  GOTOØ    EQU  171        ; Token von GOTO
                  22  RESTOREØ EQU  174        ; Token von RESTORE
                  23  GRØ      EQU  136        ; Token von GR
                  24  HGRØ     EQU  145        ; Token von HGR
                  25  HGR2Ø    EQU  144        ; Token von HGR2
                  26  *
                  27  * Zero-Page-Adressen ------
                  28  *
                  29  WNDWDTH  EQU  $21
                  30  WNDTOP   EQU  $22
                  31  WNDBTM   EQU  $23
                  32  BASL     EQU  $28
                  33  BAS2L    EQU  $2A
                  34  LINNUM   EQU  $5Ø
                  35  FRETOP   EQU  $6F
                  36  MEMSIZE  EQU  $73
                  37  CURLIN   EQU  $75
                  38  DATPNT   EQU  $7D
                  39  LOWTR    EQU  $9B
                  40  TXTPTR   EQU  $B8
                  41  HPAG     EQU  $E6
                  42  *
                  43  * Verschiedene System-Adressen -------
                  44  *
                  45  CHRGET   EQU  $ØØB1       ; Zeichen holen
                  46  SPKR     EQU  $CØ3Ø       ; Adresse des Lautsprechers
                  47  FNDLIN   EQU  $D61A       ; Zeile suchen
                  48  NEWSTT   EQU  $D7D2       ; neuen Befehl ausführen
                  49  FRMNUM   EQU  $DD67       ; Ausdruck berechnen
                  50  ERROR    EQU  $DEC9       ; Ausgabe 'Syntax Error'
                  51  GETADR   EQU  $E752       ; FAC1 --> Integer (in $5Ø,$51)
                  52  ILLQUA   EQU  $E199       ; Ausgabe 'Illegal Quant. Error'
                  53  GETNUM   EQU  $E746       ; Hole 2 Parameter aus Basic
                  54  VTABZ    EQU  $FC24
                  55  SCROLLØ  EQU  $FC7Ø
                  56  WAIT     EQU  $FCA8       ; Warte-Schleife
                  57  *
                  58  * Initialisieren des &-Vektors
                  59  *
9378: A9 93       60          LDA  #>START     ; High-Byte
937A: 8D F7 Ø3    61          STA  $Ø3F7
937D: 85 7Ø       62          STA  FRETOP+1
937F: 85 74       63          STA  MEMSIZE+1
9381: A9 9Ø       64          LDA  #<START     ; Low-Byte
9383: 85 6F       65          STA  FRETOP
9385: 85 73       66          STA  MEMSIZE
9387: 8D F6 Ø3    67          STA  $Ø3F6
938A: A9 4C       68          LDA  #$4C        ; Sprungbefehl
938C: 8D F5 Ø3    69          STA  $Ø3F5
938F: 6Ø          70          RTS
                  71  *
                  72  * War der Befehl ein 'GR'?
                  73  *
939Ø: C9 88       74  START   CMP  #GRØ        ; Mit GR-Token vergleichen
9392: DØ Ø3       75          BNE  NEXTCOM
9394: 4C BD 95    76          JMP  GR          ; GR-Routine aufrufen
9397: C9 91       77  NEXTCOM CMP  #HGRØ
9399: DØ Ø3       78          BNE  NEXTCOM1
939B: 4C CF 95    79          JMP  HGR
939E: C9 9Ø       80  NEXTCOM1 CMP #HGR2Ø
93AØ: DØ Ø3       81          BNE  NEXTCOM2
93A2: 4C E2 95    82          JMP  HGR2
93A5: C9 AB       83  NEXTCOM2 CMP #GOTOØ
93A7: FØ 3E       84          BEQ  GOTO
93A9: C9 BØ       85          CMP  #GOSUBØ
93AB: FØ 46       86          BEQ  GOSUB
```

```
93AD: C9 AE      87            CMP  #RESTOREØ
93AF: FØ 5C      88            BEQ  RESTORE
93B1: C9 9E      89            CMP  #INVERSEØ
93B3: FØ 77      9Ø            BEQ  INVERSE
93B5: C9 9F      91            CMP  #FLASHØ
93B7: DØ Ø3      92            BNE  NEXTCOM3
93B9: 4C 8B 94   93            JMP  FLASH
                 94   *
                 95   * ----- War Befehl ein 'Scroll'? -----
                 96   *
93BC: AØ ØØ      97   NEXTCOM3 LDY  #$ØØ
93BE: D9 F5 95   98   NEXTCHAR CMP  SCROLL1,Y
93C1: DØ ØE      99            BNE  SOUND?      ; Befehl <> 'Scroll'
93C3: C8         1ØØ           INY
                 1Ø1  *
                 1Ø2  * --- Alle 6 Buchstaben verglichen? ---
                 1Ø3  *
93C4: CØ Ø6      1Ø4           CPY  #$Ø6
93C6: DØ Ø3      1Ø5           BNE  NEXT
93C8: 4C 94 94   1Ø6           JMP  SCROLL      ; Routine aufrufen
93CB: 2Ø B1 ØØ   1Ø7   NEXT    JSR  CHRGET
93CE: 4C BE 93   1Ø8           JMP  NEXTCHAR
93D1: D9 FB 95   1Ø9   SOUND?  CMP  SOUND1,Y
93D4: FØ Ø3      11Ø           BEQ  NEXTC
93D6: 4C C9 DE   111           JMP  ERROR       ; 'Syntax Error'
93D9: C8         112   NEXTC   INY
93DA: CØ Ø5      113           CPY  #$Ø5        ; verglichen?
93DC: DØ Ø3      114           BNE  NEXTC1
93DE: 4C 97 95   115           JMP  SOUND
93E1: 2Ø B1 ØØ   116   NEXTC1  JSR  CHRGET
93E4: 4C D1 93   117           JMP  SOUND?
                 118  *
                 119  * ------------ GOTO(X)-Routine ---------------
                 12Ø  *
93E7: 2Ø B1 ØØ   121   GOTO    JSR  CHRGET
93EA: 2Ø 67 DD   122           JSR  FRMNUM      ; Ausdruck berechnen
93ED: 2Ø 52 E7   123           JSR  GETADR      ; und in Integer umwandeln
93FØ: 4C 41 D9   124           JMP  $D941       ; GOTO-Routine ansprechen
                 125  *
                 126  * ------------ GOSUB(X)-Routine ---------------
                 127  *
93F3: A9 Ø3      128   GOSUB   LDA  #$Ø3
93F5: 2Ø D6 D3   129           JSR  $D3D6       ; Stack prüfen
93F8: A5 B9      13Ø           LDA  TXTPTR+1    ; CHRGET-Zeiger retten
93FA: 48         131           PHA
93FB: A5 B8      132           LDA  TXTPTR
93FD: 48         133           PHA
93FE: A5 76      134           LDA  CURLIN+1    ; Zeilennr. retten
94ØØ: 48         135           PHA
94Ø1: A5 75      136           LDA  CURLIN
94Ø3: 48         137           PHA
94Ø4: A9 BØ      138           LDA  #$BØ
94Ø6: 48         139           PHA
94Ø7: 2Ø E7 93   14Ø           JSR  GOTO
94ØA: 4C D2 D7   141           JMP  NEWSTT      ; Befehl ausführen
                 142  *
                 143  * ------------ RESTORE(X)-Routine ---------------
                 144  *
94ØD: 2Ø B1 ØØ   145   RESTORE JSR  CHRGET
941Ø: 2Ø 67 DD   146           JSR  FRMNUM
9413: 2Ø 52 E7   147           JSR  GETADR
9416: 2Ø 1A D6   148           JSR  FNDLIN      ; Zeile suchen
9419: A5 9C      149           LDA  LOWTR+1     ; Adr. der Zeile
941B: 85 7E      15Ø           STA  DATPNT+1    ; in Data-Zeiger ablegen
941D: A5 9B      151           LDA  LOWTR
941F: 38         152           SEC
942Ø: E9 Ø1      153           SBC  #$Ø1
9422: 85 7D      154           STA  DATPNT
9424: C9 FF      155           CMP  #$FF
9426: FØ Ø1      156           BEQ  OVFLW
9428: 6Ø         157           RTS
9429: C6 7E      158   OVFLW   DEC  DATPNT+1
942B: 6Ø         159           RTS
                 16Ø  *
                 161  * ---------- Invertierungs-Routine ----------
                 162  *
942C: 2Ø B1 ØØ   163   INVERSE JSR  CHRGET
942F: C9 C9      164           CMP  #$C9        ; Inverse-H?
9431: DØ 29      165           BNE  INVTXT
9433: 2Ø B1 ØØ   166           JSR  CHRGET
9436: 2Ø B1 ØØ   167           JSR  CHRGET
9439: A9 ØØ      168           LDA  #$ØØ
943B: 85 Ø1      169           STA  $Ø1
943D: A2 ØØ      17Ø   NXTLINE LDX  #$ØØ
943F: AØ ØØ      171           LDY  #$ØØ
9441: A5 Ø1      172           LDA  $Ø1         ; Berechnung der
9443: 2Ø 11 F4   173           JSR  $F411       ; Adresse und Maske des
9446: AØ ØØ      174           LDY  #$ØØ        ; Punktes
```

ProDOS-Editor 1.0

Applesoft-Editor unter ProDOS-Betriebssystem

von U. Stiehl

1984, Diskette und Manual, DM 98,–
ISBN 3-7785-1024-X

Mit diesem neuen Editor – übrigens der bislang einzige deutsche ProDOS-Editor – wird dem Applesoft-Programmierer ein Werkzeug zur effektiven Programmierung unter dem Betriebssystem ProDOS gegeben, denn die früheren Editoren sind allesamt unter ProDOS nicht mehr lauffähig.

Unter anderem sind folgende Features implementiert worden:

- Zeilenorientierter Editor mit jedem erdenklichen Redigierkomfort (Insert, Delete, Tab, Restore, freie Cursorbewegung in allen vier Richtungen, Eingabe von Ctrl-Buchstaben in Applesoft-Zeilen usw.)
- Renumber (Zeilen-Umnumerierung)
- Xreference (sortierte Variablenliste)
- Suchen von Tokens, Strings und Variablen
- dezimale und hexadezimale Umrechnungen
- Ausführung von Monitorbefehlen aus dem Editor heraus
- Listen des Applesoft-Programms in speicherinterner Form als Hex-Dump
- Suchen von Hex-Folgen, Adressen oder Speicherstellen im gesamten RAM-Bereich einschließlich der Language-Card
- frei definierbare Tastatur-Macrobefehle

Der Applesoft-Editor liegt in einem von ProDOS geschützten Bereich und läßt sich per Tastendruck vorübergehend abschalten und ebenso einfach wieder aktivieren.

Gerätevoraussetzung: Apple II+, IIe oder IIc

Hüthig Software Service,
Postfach 10 28 69,
D-6900 Heidelberg

```
9448: B1 26      175  NXTBYTE  LDA  ($26),Y
944A: 49 7F      176           EOR  #$7F        ; invertieren
944C: 91 26      177           STA  ($26),Y
944E: C8         178           INY
944F: C0 28      179           CPY  #$28        ; 1 Zeile fertig?
9451: D0 F5      180           BNE  NXTBYTE
9453: E6 01      181           INC  $01         ; nächste Zeile
9455: A5 01      182           LDA  $01
9457: C9 C0      183           CMP  #$C0        ; alle Zeilen fertig?
9459: D0 E2      184           BNE  NXTLINE
945B: 60         185           RTS
                 186  *
                 187  * -------- Textseite invertieren ---------
                 188  *
945C: A2 04      189  INVTXT   LDX  #$04        ; Zeiger auf Text-
945E: 86 02      190           STX  $02         ; seite initialisieren
9460: A9 00      191           LDA  #$00
9462: 85 01      192           STA  $01
9464: A0 00      193  LOOP9    LDY  #$00
9466: B1 01      194  LOOP8    LDA  ($1),Y
9468: C9 C0      195           CMP  #$C0
946A: 90 0D      196           BCC  LOOP10
946C: 49 C0      197  LOOP13   EOR  #$C0
946E: 91 01      198  LOOP11   STA  ($1),Y
9470: 88         199           DEY
9471: D0 F3      200           BNE  LOOP8
9473: E6 02      201           INC  $02
9475: CA         202           DEX
9476: D0 EC      203           BNE  LOOP9
9478: 60         204           RTS
9479: C9 A0      205  LOOP10   CMP  #$A0
947B: 90 05      206           BCC  LOOP20
947D: 49 80      207           EOR  #$80
947F: 4C 6E 94   208           JMP  LOOP11
9482: C9 20      209  LOOP20   CMP  #$20
9484: 90 E6      210           BCC  LOOP13
9486: 49 80      211           EOR  #$80
9488: 4C 6E 94   212           JMP  LOOP11
                 213  *
                 214  * ------------- Flash-Routine ----------------
                 215  *
948B: 20 B1 00   216  FLASH    JSR  CHRGET
948E: 20 5C 94   217           JSR  INVTXT      ; FLASHen durch
9491: 4C 5C 94   218           JMP  INVTXT      ; 2 x invertieren
                 219  *
                 220  * ------------- Scroll-Routine ---------------
                 221  *
9494: A9 A0      222  SCROLL   LDA  #$A0
9496: 85 FF      223           STA  $FF
9498: 20 B1 00   224           JSR  CHRGET
949B: 20 B1 00   225           JSR  CHRGET
949E: C9 88      226           CMP  #136        ; Scroll-GR?
94A0: D0 03      227           BNE  NIXGR
94A2: 4C 72 95   228           JMP  SCRGR
94A5: C9 47      229  NIXGR    CMP  #'G         ; Grafik-Scroll?
94A7: D0 03      230           BNE  NOGRAFIK
94A9: 4C 53 95   231           JMP  SCROLLG
94AC: C9 52      232  NOGRAFIK CMP  #'R
94AE: F0 15      233           BEQ  RECHTS
94B0: C9 4C      234           CMP  #'L
94B2: F0 37      235           BEQ  LINKS
94B4: C9 55      236           CMP  #'U
94B6: D0 03      237           BNE  LOOP31
94B8: 4C 4D 95   238           JMP  UP
94BB: C9 44      239  LOOP31   CMP  #'D
94BD: D0 03      240           BNE  FEHLER
94BF: 4C 17 95   241           JMP  DOWN
94C2: 4C C9 DE   242  FEHLER   JMP  ERROR
                 243  *
                 244  * --- Rechts-Scrollen -----
                 245  * Umgeschriebene Upscroll-
                 246  * Routine vom Autostart-Monitor
                 247  *
94C5: A5 23      248  RECHTS   LDA  WNDBTM
94C7: 10 17      249           BPL  LOOP15
94C9: 85 01      250  LOOP17   STA  1
94CB: 20 24 FC   251           JSR  VTABZ
94CE: A4 21      252           LDY  WNDWDTH
94D0: 88         253           DEY
94D1: 88         254  LOOP16   DEY
94D2: B1 28      255           LDA  (BASL),Y
94D4: C8         256           INY
94D5: 91 28      257           STA  (BASL),Y
94D7: 88         258           DEY
94D8: D0 F7      259           BNE  LOOP16
94DA: A5 FF      260           LDA  $FF
94DC: 91 28      261           STA  (BASL),Y
94DE: A5 01      262           LDA  $01
```


DB-MEISTER

Adreß- und Schemabriefprogramm

Der DB-Meister ist ein in Assembler geschriebenes, ungewöhnlich schnelles, unkompliziertes und zugleich „narrensicheres" Adreß-, Datei- und Schemabriefprogramm.

Technische Daten

- Recordlänge bis zu 230 Zeichen
- 560 bis 1000 Records pro Datendiskette
- Maximal 25 Felder pro Record
- Suche nach 3 Indexfeldern
- Ausdruck der Dateien als Etiketten, Listen und Schemabriefe (mit Felder- und Tastatureinschüben an beliebigen Stellen des Formbriefes)
- normal kopierbare Programmdiskette, unterteilt in Hauptprogramme und diverse Hilfsprogramme
- einsatzfähig auf Apple IIe, IIc oder II Plus mit 2 Drives (1 Drive ebenfalls möglich)

Gesamtpreis 290,– (2 Disketten + gedrucktes Manual)

U. Stiehl

c/o Dr. A. Hüthig Verlag
Postfach 10 28 69 · 6900 Heidelberg

AS.ERWEITERUNG und ProDOS

Damit die AS.ERWEITERUNG auch unter ProDOS läuft, wurde das Assemblerprogramm mit ORG $9778 (statt $9378) neu assembliert, nachdem die HIMEM-Pokes (MEMSIZ) zu Beginn des Programms durch die entsprechende Anzahl von NOPs ersetzt wurden. Diese Neuversion findet sich unter dem Namen „AS.ERW-.PRO" auf der Peeker-Sammeldisk #6 und ist – wenn man von der Verschiebung um $0400 nach oben absieht – mit der DOS-Version identisch. Ferner war ein kurzes Applesoft-Start-Programm namens „AS.ERW.PROSTART" erforderlich, das HIMEM über die sog. GETBUFFER-Routine ändert. Kopieren Sie also AS.ERW-.PRO und AS.ERW.PROSTART von der DOS-Sammeldisk auf Ihre ProDOS-Disk (mit Hilfe der CONVERT-Utility) und starten Sie dann die Ampersand-Routinen mit RUN AS.ERW.PRO

Die Speicherverteilung sieht dann unter ProDOS wie folgt aus:

$9300: HIMEM
$9300-$96FF: ProDOS-Puffer
$9778-$99FF: AS.ERW.PRO
$9A00: Beginn von BASIC.SYSTEM

Wie man unter ProDOS Assemblerprogramme oberhalb von HIMEM ansiedelt, wird in einem späteren Peeker-Artikel gezeigt. U. Stiehl

```
94E0: 38          263  LOOP15   SEC
94E1: E9 01       264            SBC  #$01
94E3: 38          265            SEC
94E4: C5 22       266            CMP  WNDTOP
94E6: 10 E1       267            BPL  LOOP17
94E8: 4C B1 00    268            JMP  CHRGET
                  269  *
                  270  * --- Links-Scrollen ------
                  271  * Umgeschriebene Upscroll-Routine
                  272  * vom Autostart-Monitor
                  273  *
94EB: C6 21       274  LINKS    DEC  WNDWDTH
94ED: A5 23       275            LDA  WNDBTM
94EF: 10 18       276            BPL  LOOP21
94F1: 85 01       277  LOOP22   STA  1
94F3: 20 24 FC    278            JSR  VTABZ
94F6: A0 00       279            LDY  #$00
94F8: C8          280  LOOP23   INY
94F9: B1 28       281            LDA  (BASL),Y
94FB: 88          282            DEY
94FC: 91 28       283            STA  (BASL),Y
94FE: C8          284            INY
94FF: C4 21       285            CPY  WNDWDTH
9501: D0 F5       286            BNE  LOOP23
9503: A5 FF       287            LDA  $FF
9505: 91 28       288            STA  (BASL),Y
9507: A5 01       289            LDA  $01
9509: 38          290  LOOP21   SEC
950A: E9 01       291            SBC  #$01
950C: 38          292            SEC
950D: C5 22       293            CMP  WNDTOP
950F: 10 E0       294            BPL  LOOP22
9511: 20 B1 00    295            JSR  CHRGET
9514: E6 21       296            INC  WNDWDTH
9516: 60          297            RTS
                  298  *
                  299  * ----- Down-Scroll ------
                  300  * Umgeschriebene Upscroll-Routine
                  301  * vom Autostart-Monitor
                  302  *
9517: A6 23       303  DOWN     LDX  WNDBTM
9519: CA          304            DEX
951A: 8A          305            TXA
951B: 48          306            PHA
951C: 20 24 FC    307            JSR  VTABZ
951F: A5 28       308  LOOP40   LDA  BASL
9521: 85 2A       309            STA  BAS2L
9523: A5 29       310            LDA  BASL+1
9525: 85 2B       311            STA  BAS2L+1
9527: A4 21       312            LDY  WNDWDTH
9529: 88          313            DEY
952A: 68          314            PLA
952B: 38          315            SEC
952C: E9 01       316            SBC  #$01
952E: 38          317            SEC
952F: C5 22       318            CMP  WNDTOP
9531: 30 0D       319            BMI  LOOP41
9533: 48          320            PHA
9534: 20 24 FC    321            JSR  VTABZ
9537: B1 28       322  LOOP42   LDA  (BASL),Y
9539: 91 2A       323            STA  (BAS2L),Y
953B: 88          324            DEY
953C: 10 F9       325            BPL  LOOP42
953E: 30 DF       326            BMI  LOOP40
9540: A4 21       327  LOOP41   LDY  WNDWDTH
9542: 88          328            DEY
9543: A5 FF       329            LDA  $FF
9545: 91 28       330  LOOP45   STA  (BASL),Y
9547: 88          331            DEY
9548: 10 FB       332            BPL  LOOP45
954A: 4C B1 00    333            JMP  CHRGET
                  334  *
                  335  * ---- Up-Scroll -------
                  336  *
954D: 20 70 FC    337  UP       JSR  SCROLL0     ; Upscroll
9550: 4C B1 00    338            JMP  CHRGET
                  339  *
                  340  * --- Low-Grafik-Scroll ----
                  341  *
9553: A9 00       342  SCROLLG  LDA  #$00
9555: 85 FF       343            STA  $FF
9557: 85 22       344            STA  WNDTOP
9559: 85 23       345            STA  WNDBTM
955B: 48          346            PHA
955C: A9 14       347            LDA  #20
955E: 85 23       348            STA  WNDBTM
9560: 20 B1 00    349            JSR  CHRGET
9563: C9 55       350            CMP  #'U         ; Up?
```

```
9565: F0 1E       351            BEQ  UPSCR
9567: 20 AC 94    352            JSR  NOGRAFIK
956A: A9 14       353  WDWRESTO LDA  #20
956C: 85 22       354            STA  WNDTOP
956E: 68          355            PLA
956F: 85 23       356            STA  WNDBTM
9571: 60          357            RTS
9572: A9 00       358  SCRGR    LDA  #$00
9574: 85 22       359            STA  WNDTOP
9576: 85 FF       360            STA  $FF
9578: A5 23       361            LDA  WNDBTM
957A: 48          362            PHA
957B: A9 14       363            LDA  #20
957D: 85 23       364            STA  WNDBTM
957F: 20 C5 94    365            JSR  RECHTS
9582: 4C 6A 95    366            JMP  WDWRESTO
9585: 20 70 FC    367  UPSCR    JSR  SCROLL0
9588: 20 B1 00    368            JSR  CHRGET
958B: A0 27       369            LDY  #$27
958D: A9 00       370            LDA  #$00
958F: 99 D0 05    371  INC      STA  $05D0,Y
9592: 88          372            DEY
9593: 10 FA       373            BPL  INC
9595: 30 D3       374            BMI  WDWRESTO
                  375  *
                  376  * ---------- Sound-Routine ----------
                  377  *
9597: 20 B1 00    378  SOUND    JSR  CHRGET
959A: 20 46 E7    379            JSR  GETNUM      ; Lese Parameter
959D: A4 51       380            LDY  LINNUM+1
959F: F0 03       381            BEQ  OK
95A1: 4C 99 E1    382            JMP  ILLQUA
95A4: 86 01       383  OK       STX  $01
95A6: A4 01       384  REPEAT   LDY  $01
95A8: A6 50       385  REP1     LDX  LINNUM
95AA: A9 04       386            LDA  #$04
95AC: 20 A8 FC    387            JSR  WAIT        ; Wait-Routine
95AF: AD 30 C0    388            LDA  SPKR        ; Lautsprecher
95B2: E8          389  DELAY    INX
95B3: D0 FD       390            BNE  DELAY
95B5: 88          391            DEY
95B6: D0 F0       392            BNE  REP1
95B8: C6 01       393            DEC  $01
95BA: D0 EA       394            BNE  REPEAT
95BC: 60          395            RTS
                  396  *
                  397  * ---------- &GR-Routine ----------
                  398  *
95BD: AD 50 C0    399  GR       LDA  $C050
95C0: AD 53 C0    400            LDA  $C053
95C3: AD 54 C0    401            LDA  $C054
95C6: AD 56 C0    402            LDA  $C056
95C9: 20 B1 00    403            JSR  CHRGET
95CC: 4C 49 FB    404            JMP  $FB49
                  405  *
                  406  * ---------- &HGR-Routine ----------
                  407  *
95CF: AD 50 C0    408  HGR      LDA  $C050
95D2: AD 53 C0    409            LDA  $C053
95D5: AD 54 C0    410            LDA  $C054
95D8: AD 57 C0    411            LDA  $C057
95DB: A9 20       412            LDA  #$20
95DD: 85 E6       413            STA  HPAG        ;          1. Seite
95DF: 4C B1 00    414            JMP  CHRGET
                  415  *
                  416  * ---------- &HGR2-Routine ----------
                  417  *
95E2: AD 50 C0    418  HGR2     LDA  $C050
95E5: AD 52 C0    419            LDA  $C052
95E8: AD 55 C0    420            LDA  $C055
95EB: AD 57 C0    421            LDA  $C057
95EE: A9 40       422            LDA  #$40
95F0: 85 E6       423            STA  HPAG        ;          2. Seite
95F2: 4C B1 00    424            JMP  CHRGET
                  425  *
                  426  * Ablegen von 'Scroll' und
                  427  * 'Sound' als ASCII-Code zum
                  428  * späteren Vergleich
                  429  *
95F5: 53 43 52    430  SCROLL1  ASC  'SCROLL'
95F8: 4F 4C 4C
95FB: 53 4F 55    431  SOUND1   ASC  'SOUND'
95FE: 4E 44
```

648 Bytes

RAM-Disk-Driver für Pascal 1.1

von Michael Schröter

1. Theorie

Eine ausgesprochen nützliche Erweiterung für den Apple IIe ist die erweiterte 80-Zeichenkarte. Eine der vielen neuen Möglichkeiten, den zusätzlichen Speicher auszunutzen, besteht in der Simulation einer Diskette. Dies ist natürlich vom verwendeten Betriebssystem abhängig. Nachdem eine solche Simulation für DOS 3.3 bereits veröffentlicht wurde (vgl. (1) sowie Peeker 1/2, 85 und 4/85), soll hier ein entsprechendes Konzept für PASCAL 1.1 vorgestellt werden. Zunächst einige Vorbemerkungen:

Im Gegensatz zu DOS 3.3 werden Peripheriegeräte (Devices oder Volumes) in Pascal nicht über Slot und Drive selektiert, sondern über ihre entsprechende Unitnummer angesprochen (vgl. (2), S. 26, 276-277). Die Verwaltung übernimmt der Interpreter, die eigentliche Ausführung das BIOS (Basic Input Output System). Schnittstelle zwischen dem p-code (also dem compilierten Pascal-Programm) und dem BIOS sind die Unit-Prozeduren des Interpreters. Da (3) nicht alle verfügbaren Prozeduren beschreibt, seien sie hier noch einmal kurz genannt:

UNITBUSY (unitno), UNITWAIT (unitno), UNITCLEAR (unitno), UNITSTATUS (unitno, array, mode);
UNITREAD (unitno, array, length, block, mode);
UNITWRITE (unitno, array, length, block, mode).

Bei genauerer Untersuchung stellt man fest, daß selbst „höhere" Prozeduren wie BLOCKREAD und sogar GET, PUT, SEEK, WRITE usw. ausschließlich über diese Unit-Prozeduren mit dem BIOS verkehren. Weiterhin sind nur die 4 letztgenannten Prozeduren unitspezifisch, denn nur sie werten die Unit-Number auch tatsächlich aus, um die entsprechenden BIOS-Routinen anzusteuern. Den eigentlichen Aufruf übernimmt dann die Schnittstelle zwischen Interpreter und BIOS.

Hier werden die einzelnen Devices in verschiedenen Tabellen verwaltet. Die erste Tabelle ($FE80..$FEAF) enthält die „benutzer-definierbaren Sprungvektoren". Der Anwender kann bis zu 16 eigene Devices verwalten lassen, die dann den Unitnummern 128..143 zugeordnet sind (übrigens eine Möglichkeit, über die sich die Apple-Handbücher ebenfalls ausschweigen). Da das Laufzeitsystem hiervon keinen Gebrauch macht, soll darauf auch nicht näher eingegangen werden.

Die systemspezifischen Geräte (also die Units 1..12) werden in einer eigenen Tabelle ($FEB0..$FEC7) verwaltet. Für jedes Device sind dabei 2 Bytes vorgesehen, wobei bei einem Aufruf folgende Zuordnung erfolgt: Ist das High-Byte (also $FEB1 für Device 1, $FEB3 für Device 2 usw.) der Tabelle = $00, so wird die BIOS-Routine zur Disk-Verwaltung gerufen. Ist das High-Byte hingegen = $FF, so liegt ein Non-blocked-Device vor (also etwa CONSOLE oder PRINTER), und eine dritte Tabelle ($EEAD..$EECC) steuert den BIOS-Aufruf. Alle anderen vorgefundenen Werte werden als eine „indexed indirect address" behandelt, d.h. als Anfangsadresse-1 der Handler-Routine. Genau hier hakt auch das folgende Programm ein. Noch ein Wort zur Übergabe der Parameter: Wird der Interpreter für eine Diskettenoperation verlassen, so findet das BIOS die Parameter in folgender Ordnung auf dem Systemstack (je 2 Bytes mit dem Low-Byte an oberster Stelle):

Top of Stack:
Blocknummer
Anzahl der zu übertragenen Bytes
Adresse des Arrays
Unitnummer
Modeparameter

Bei Diskettenoperationen ist der Modeparameter immer = $00. Das gleiche gilt für das High-Byte der Unitnummer.

2. Praxis

Da der 6502 lediglich in der Lage ist, 64K zu adressieren, muß man beim Verwalten der zusätzlichen 64K auf die Softswitches bzw. die Firmware-Routinen AUXMOVE und XFER zurückgreifen (vgl. (4), S. 71). Diese beiden Prozeduren übernehmen auch die Hauptarbeit im Programm. Allerdings ist man mit ihnen nur in der Lage, den Speicherbereich von $0200 bis $BFFF zu manipulieren. (Anm. d. Red.: Die gleiche Einschränkung gilt auch für das in (1) gelistete Programm. Deshalb soll zu einem späteren Zeitpunkt im Peeker ein Pascal-RAM-Disk-Driver veröffentlicht werden, der wie die Peeker-RAM-Disk-Programme für CP/M und DOS 3.3 die 64K voll ausnutzt. Dann muß man allerdings auf AUXMOVE teilweise verzichten.)
In dem Bereich $0200-$BFFF wird die RAM-Disk angelegt. Bekanntlich organisiert Pascal seine Disketten im Blockformat (1 Block = 512 Bytes). Ein File wird dann in zusammenhängenden, hintereinanderliegenden Blöcken gespeichert. Bei einer neu initialisierten Diskette werden die eigentlichen Files erst ab Block 6 gespeichert, weil die Blöcke 0 und 1 für das Boot-Programm reserviert sind und die Blöcke 2, 3, 4 und 5 das Directory der Diskette enthalten. Die RAM-Disk beginnt mit Block 2, da die Boot-Blöcke bei einer RAM-Disk keinen Sinn ergeben. Zu beachten ist nun ferner, daß der Bereich $0400..$07FF für die 80-Zeichendarstellung benutzt wird. Somit beginnt Block 2 bei $0800 und erstreckt sich bis einschließlich $09FF (was genau 512 Bytes ausmacht). Es kommt nun nur darauf an, die Blocknummer in die entsprechende Anfangsadresse des Hilfsspeichers zu transformieren. Das besorgt folgende Umrechnung:

Adresse (high):= Blocknummer * 2 + 4
Adresse (low):= 0

Die maximale Adresse für einen Block ist $BE00. Mit $BE = 190 hat man also als

höchste Blocknummer: (190–4) / 2 = 93 – ein Gewinn von 91 zusätzlichen Blöcken – dies sind immerhin 32% einer normalen Diskette. Schematisch ergibt sich dann folgender Aufbau für die RAM-Disk:

$C000 RAM-Top
$BE00 Block 93
$BC00 Block 92
usw.
$0A00 Block 3
$0800 Block 2
$0400 Text-Page 1X
$0200 RAM-Disk-Driver
$0000 RAM-Bottom

An das Progamm selbst wurde die Bedingung gestellt, daß es von allen Systemteilen aus verfügbar sein sollte, so daß auch Filer, Compiler usw. zugreifen können. Es muß daher an einem Platz implementiert sein, an dem es nicht überschrieben werden kann. Der Interpreter selbst läßt für derartige Zusätze leider keinen Platz. Der Laufzeitkeller und die Halde verbieten, das Programm im Programmstack unterzubringen. Ein geeignetes „Plätzchen" findet sich allein im ungenutzten Bereich der Workarea des BIOS ab $BF80. Dort läßt sich aber beim besten Willen kein ganzes Treiberprogramm unterbringen. Somit wurde der Treiber, der für Parametertest, Umrechnung und Datentransfer zuständig ist, in den Hilfsspeicher „ausquartiert", und zwar ab $0200. Im Hauptspeicher ab $BF80 liegt lediglich ein kurzes Interface, das alle Vorbereitungen für XFER trifft, den Treiber über XFER anspringt und schließlich alle Spuren beseitigt.
Ist das Programm erst einmal installiert, so muß nur noch die vorgenannte Tabelle auf $BF80 justiert werden, und die RAM-Disk ist angeschlossen. Die Strukturierung und den endgültigen Einbau in das Laufzeitsystem übernimmt dann die sonst wenig nützliche Zap-Funktion des Filers.

3. Programm

Das Programm gliedert sich in drei Teile:

1. Das Pascal-Programm **RAMDISK94**:
Dieses Host-Programm übernimmt den Aufruf von INIT und prüft, ob die RAM-Disk schon initialisiert wurde. Der Hilfsspeicher wird nur bei Stromabschaltung gelöscht, so daß der Filer lediglich bei einem Kaltstart benötigt wird. Die Konstante Device ist die Volume-Nummer der RAM-Disk. Hier vorgenommene Änderungen müssen sinngemäß auch bei INIT und INSTALL erfolgen.

2. Das Assembler-Programm **INIT**:
Dieses Programm umfaßt die Initialisierungsroutine, die den Treiber in den Hilfsspeicher nach $0200 und das Interface in den Hauptspeicher nach $BF80 schiebt und schließlich die Device-Tabelle anpaßt. Ferner enthält INIT die Interface-Routine (INFCE) und den Treiber selbst (MANAGE).

3. Den Exec-File **INSTALL**:
Es dient der Filer-Steuerung beim Kaltstart. Man erhält den File folgendermaßen: Aus der Command-Prompt-Line heraus wähle man
M(ake exec.

Zu diesem Zeitpunkt müssen alle betroffenen Files verfügbar sein, also insbesondere der Filer selbst. Als Beispiel wird hier alles auf die Volume-Nummer 4 bezogen. Dann entwickelt sich folgender Dialog:

New exec name: #4:INSTALL.TEXT
Terminator = %, change it? N <hier beginnt der Exec-File
F <um den Filer aufzurufen
Z <das Z(ap Kommando
#11 <die installierte Device-Nummer
N <keine Directory-Kopie
94 <Anzahl der Blöcke auf Device #11 (s.o.)
RAMDISK <Device-Name
Y <RAMDISK ist der richtige Name

(Der folgende Dialog dient zur Übertragung einiger häufig gebrauchter Files auf die neue „Diskette". Man kann natürlich auch andere Programme übertragen oder sie ganz frei lassen)

T <für T(ransfer
#4:SYSTEM.FILER <z.B. den Filer
#11:$ <zur neuen „Diskette"
#4:SYSTEM.EDITOR <und den Editor
#11:$
Q <das war's
%% <Ende EXEC File

Damit ist die RAM-Disk verfügbar.

Literatur

(1) Douglas Peterson, RAMDISK64, Nibble Magazine, 9/1984, S. 33.
(2) Apple Computer, Apple Pascal Operating System Reference Manual.
(3) Apple Computer, Apple Pascal Language Reference Manual.
(4) Apple Computer, Reference Manual for IIe only.

Peeker-Sammeldisk #6

zu Heft 6/1985, einzeln DM 28,–
im Fortsetzungsbezug DM 20,–

A 024 HELLO
B 114 ASMDIV

B 002 CURSOR1
T 004 T.CURSOR1
B 002 CURSOR2
T 004 T.CURSOR2
B 002 LINIE
T 005 T.LINIE
B 002 VIERECK
T 005 T.VIERECK
B 002 BOX
T 009 T.BOX
B 002 HINTERGRUND
T 005 T.HINTERGRUND
B 002 PAGE.SWAP
T 004 T.PAGE.SWAP

A 003 WANDERNDER.STRICH
A 004 KOMPRESSOR.DEMO
A 003 KREIS.1
A 004 KREIS.2
A 004 KREIS.3
B 002 FLIPPER
T 005 T.FLIPPER
B 002 KOMPRESSOR
T 012 T.KOMPRESSOR

B 003 OLYMPIA
T 021 T.OLYMPIA

A 043 FOURIER.MAIN
A 019 FOURIER.SYN
A 020 FOURIER.SPEC

B 004 AS.ERWEITERUNG
T 028 T.AS.ERWEITERUNG
A 003 AS.ERW.PROSTART
B 004 AS.ERW.PRO
T 017 T.AS.ERW.PRO

T 002 INSTALL.PASCAL.SOURCE
T 005 RAMDISK94.PASCAL.SOURCE
T 032 INIT.PASCAL.SOURCE

A 002 RAMDISK.INIT.DOS
B 002 AUXDRIVER
T 010 T.AUXDRIVER
B 002 MOVEDRIVER
T 004 T.MOVEDRIVER
B 002 RAMDISK.FORMATTER
T 006 T.RAMDISK.FORMATTER

A 002 SOLITAIRE.START
A 029 SOLITAIRE
B 004 SOLITAIRE.B
T 016 T.SOLITAIRE.B

Hüthig Software Service
Postfach 10 28 69 · 6900 Heidelberg

RAMDISK94.PASCAL.SOURCE

```
PROGRAM RAMDISK94;

USES CHAINSTUFF;

CONST
  device = 11;

PROCEDURE init; EXTERNAL;

FUNCTION check : boolean;
{check stellt fest, ob RAMDISK schon vorhanden ist}
{Falls true, braucht nicht mehr gezapt zu werden}
VAR
  idrec :       {idrec ist ein Ausschnitt aus dem}
    RECORD      {eventuellen Directory-Eintrag}
      first  : integer;
      last   : integer;
      ftype  : Ø..8;
      vname  : string[7]
    END;
BEGIN
  check := false;
{$I-}
  UNITREAD (device,idrec,SIZEOF(idrec),2);
{$I+}
  IF IORESULT = Ø THEN
    WITH idrec DO BEGIN
      IF (first=Ø) AND (last=6) AND (ftype=Ø)
        AND (vname='RAMDISK') THEN
          check := true {RAM-Disk schon initialisiert}
    END
END;

BEGIN
  SETCHAIN (''); INIT;
  IF NOT check THEN
    SETCHAIN ('EXEC/INSTALL')
END.
```

INIT.PASCAL.SOURCE

```
; Installationsroutine
;-----------------------------------------
A1              .EQU 3C
A2              .EQU 3E
A4              .EQU 42
XADR            .EQU 3ED

RETURN          .EQU Ø

XFER            .EQU ØC314
AUXMOVE         .EQU ØC311

SLOTCXRD        .EQU ØCØ15
SLOTCXON        .EQU ØCØØ6
SLOTCXOF        .EQU ØCØØ7

HIRESOF         .EQU ØCØ56

DISKSTAT        .EQU ØDCC5      ; Default-Disk-Status-Routine

DSKVCT          .EQU ØFEAE      ; Basisadresse der Unittabelle
;-----------------------------------------
                .MACRO POP
                PLA
                STA %1
                PLA
                STA %1+1
                .ENDM

                .MACRO PUSH
                LDA %1+1
                PHA
                LDA %1
                PHA
                .ENDM

                .PROC INIT
;-----------------------------------------
;
; Die Initialisierungsroutine
;
;-----------------------------------------
                POP RETURN              ; Platz schaffen
                LOAD A1,RETURN+2,8
                LDA #ØØ                 ; Treiber im
                STA A4                  ; Hilfsspeicher
                LDA #Ø2                 ; ab $Ø2ØØ ablegen
                STA A4+1
                LDA DRVST
                STA A1
                LDA DRVST+1
                STA A1+1
                LDA DRVED
                STA A2
                LDA DRVED+1
                STA A2+1
                JSR MVE                 ; verschieben
                LOAD INFCE,ØBF8Ø,7F     ; Interface nach
                                        ; $BF8Ø schieben.
                LDA DEVICE              ; Tabelle für
                ASL A                   ; Device #11 justieren
                TAY
                LDA #7F
                STA DSKVCT,Y
                INY
                LDA #ØBF
                STA DSKVCT,Y
                LOAD RETURN+2,A1,8      ; Rekonfiguration
                PUSH RETURN
                RTS                     ; und Ende

MVE             LDA SLOTCXRD            ; Firmware-Status
                STA STAT                ; testen und sicher-
                STA SLOTCXOF            ; stellen, daß
                SEC                     ; Firmware zugeschaltet
                JSR AUXMOVE             ; ist.
                LDA STAT                ; Firmware-Status wieder
                BMI BCK                 ; herstellen.
                STA SLOTCXON
BCK             RTS

STAT            .BYTE
DEVICE          .BYTE 11.               ; kann geändert werden
                \
;-----------------------------------------
;
; Das Interface Programm zur Treibersteuerung
;
; X- Register enthält Operationsparameter:
;
;       X = Ø  :  UNITREAD
;       X = 1  :  UNITWRITE
;       X = 2  :  UNITCLEAR
;       X = 3  :  UNITSTATUS
;-----------------------------------------
INFCE           CPX #2          ; keine Aktivität bei
                BNE DEFSTAT     ; UNITCLEAR
                LDX #ØØ
                RTS

DEFSTAT         CPX #3          ; Stack leeren durch
                BNE BEGIN       ; Default-UNITSTATUS
                JMP DISKSTAT

BEGIN           STA HIRESOF     ; Wichtig: Abschalten, um
                                ; Überschreibungen zu verhindern
                POP RETURN      ; alte Werte sichern
                LDA XADR
                STA RETURN+1Ø.
                LDA XADR+1
                STA RETURN+11.

                LDA #ØØ         ; $Ø2ØØ ist Treiberstart
                STA XADR
                LDA #Ø2
                STA XADR+1

                LDA SLOTCXRD    ; s.o., Firmware muß zugeschaltet
                STA RETURN+12.  ; sein
                STA SLOTCXOF
                CLV
                SEC
```

```
                JSR XFER          ; JSR wird verwendet, um die
                                  ; Return-Adresse zu sichern

                LDA RETURN+12.    ; <- Return-Punkt ist hier
                BMI DONE
                STA SLOTCXON      ; Firmware wieder herstellen
DONE            LDA RETURN+1Ø.
                STA XADR          ; die alten Werte wieder einsetzen
                LDA RETURN+11.
                STA XADR+1
                PUSH RETURN
                RTS               ; zurück zum Hauptprogramm


;-----------------------------------------
;
; Es folgt das Treiber Programm.
; Zuoberst auf dem Stack liegt die
; Return-Adresse-1
;
;-----------------------------------------

DRVST           .WORD DRVST+2
;-----------------------------------------
MANAGE          CLC               ; Return-Adresse fixieren
                PLA               ; und Workarea freimachen
                ADC #Ø1
                STA XADR
                PLA
                ADC #ØØ
                STA XADR+1
                LOAD A1,RETURN+2,8
                PLA               ; Blocknummer holen
                TAY
                PLA
                CPY #94.          ; Block >= 94 (dez)
                BCS ERROR         ; oder < 2 ist unzulässig
                CPY #2
                BCC ERROR
                TYA               ; Umrechnung auf Speicheradresse
                ASL A
                CLC
                ADC #4
                TAY
                TXA               ; verzweigen zur entsprechenden
                BEQ READ          ; Operation

WRITE           STY A4+1          ; UNITWRITE simulieren
                LDA #Ø
                STA A4
                PLA
                STA A2
                PLA
                STA A2+1
                CLC
                PLA
                STA A1
                ADC A2
                STA A2
                PLA
                STA A1+1
                ADC A2+1
                STA A2+1
                SEC               ; C=1 für Main-Aux-Transfer
                BCS DISCARD
```

```
READ            STY A1+1          ; UNITREAD simulieren
                LDA #Ø
                STA A1
                CLC
                PLA
                ADC A1
                STA A2
                PLA
                ADC A1+1
                STA A2+1
                PLA
                STA A4
                PLA
                STA A4+1
                CLC               ; C=Ø für Aux-Main-Transfer
DISCARD         PLA               ; Unitnummer und
                PLA
                PLA               ; Modeparameter vergessen
                PLA
                PHP

                LDA A2            ; End-Adresse von Source
                BNE EXEC          ; ausgleichen
                DEC A2+1
EXEC            DEC A2
                PLP
                JSR AUXMOVE       ; Ausführung und
OUT             LDX #Ø            ; Rückkehr ohne Fehler
                BEQ BACK

ERROR           PLA               ; Im Falle eines Fehlers:
                PLA
                PLA               ; Stack leeren und
                PLA               ; mit Fehlerindikator
                PLA               ; 'No room on volume'
                PLA               ; zurück
                PLA
                PLA
                LDX #8

BACK            LOAD RETURN+2,A1,8  ; alte Werte wieder
OUT1            CLC                 ; einsetzen und zurück
                CLV                 ; zum Interface
                JMP XFER
;-----------------------------------------
DRVED           .WORD DRVED-1

                .END
```

INSTALL.PASCAL.SOURCE

```
%FZ#11
N94
RAMDSIISK
YT#4:SYSTEM.FILER
#11:$
T#4:SYSTEM.EDITOR
#11:$
Q%%%%
```

Inserentenverzeichnis peeker 6/85

Seite

Entwicklungsbüro Brecht, Fellbach-Schmiden 17
Bühler Elektronik, Baden-Baden 68
ccp datentechnik, Hamburg 32
Computerladen Memmingen, Memmingen 71
copy-team gmbh, Erlagen 19
D.O.S. Computersysteme, Schwäbisch Hall 17
Erbrecht Computer Related Products, Hamburg 10
erphi electronic GmbH, Großhelfendorf . . . . . 4. US
hib Computerladen, Nürnberg 19
IBS Computertechnik, Bielefeld . . . . . 2. US
Interkom Electronic, Isenhagen 76
Intus, Waldshut-Tiengen 32
Klaus Jeschke, Kelkheim 8
E.-W. Meyer, Frohnhausen 8
Micromint Computer GmbH, Erkrath 15
Ulf Mohwinkel Electronic, Leverkusen 17
pandabooks, Berlin 9
pandasoft, Berlin 13
r + r electronic, Heidelberg 35
Schappach Computer, Mannheim 32
Tewi Verlag, München . . . . . 3. US
Tombstone-Micro, Berlin 15
Ueding electronics, Menden 15

ProDOS

Theorie und Praxis

von Ulrich Stiehl

Im Peeker können wir aus Platzgründen nicht alle Feinheiten des komplizierten Betriebssystems ProDOS darstellen. Deshalb erscheinen im Hüthig-Verlag in einigen Wochen zwei neue ProDOS-Bücher, das eine mehr theoretische Werk von Arne Schäpers, das andere mehr praktische Buch von mir selbst. Hierzu sind einige grundsätzliche Anmerkungen erforderlich.

ProDOS für Aufsteiger, Band 2
von U. Stiehl, Juni 1985, ca. 180 Seiten, DM 28,-
(Begleitdiskette gesondert erhältlich, ebenfalls DM 28,-)

Im Gegensatz zu DOS 3.3 hat die Firma Apple bei ProDOS eine neue Politik eingeschlagen. Um einerseits die Besitzer von Nachbauten (einsichtig!) und andererseits die Produzenten von Fremdlaufwerken (weniger einsichtig!) von der Benutzung von ProDOS auszuschließen, erscheinen in fast regelmäßigen Abständen neue ProDOS-Versionen. Dies war zu dem Zeitpunkt, als ich den ersten Band von „ProDOS für Aufsteiger" verfaßte (Mai 1984), noch nicht abzusehen. Ursprünglich sollte der zweite Band neben einer Einführung in die Applesoft-Programmierung unter ProDOS eine Fülle von Tips und Tricks enthalten, um beispielsweise fremde Laufwerke auf ProDOS anzupassen usw. Neben den Urversionen für Entwickler sind jedoch inzwischen mindestens 4 Versionen aufgetaucht, nämlich ProDOS 1.0.1, 1.0.2, 1.1 und 1.1.1, und weitere Versionen werden mit Sicherheit folgen. Programme, die ProDOS an *absoluten* Adressen patchen, wie beispielsweise das FORMAT.LC-Programm im Peeker, Heft 5/85, sind damit versionsabhängig und in keinem Fall portabel. Selbst bei Diversi-DOS und anderen DOS-3.3-Varianten konnte man sich darauf verlassen, daß sich grundlegende Systemroutinen wie die RWTS immer an derselben Stelle befanden. Begann beispielsweise der ProDOS-Disk-Driver bis zur Version 1.0.2 noch bei $F800, wie auch im „ProDOS Technical Reference Manual" der Firma Apple angegeben, so beginnt er bei der Version 1.1.1 jetzt bei $D000. Aufgrund dieser geänderten Sachlage habe ich für den zweiten Band von „ProDOS für Aufsteiger" *alle* Programme so „seriös" geschrieben, daß sie grundsätzlich auf allen bisherigen und möglicherweise auch auf allen zukünftigen Versionen laufen, es sei denn, daß die MLI-Schnittstelle und die Dateistrukturen als solche radikal geändert werden.

Der zweite Band beginnt mit einer bewußt leichtverständlichen Einführung in das BASIC.SYSTEM für Applesoft-Programmierer. Danach werden vermischte Tips und Tricks gegeben, soweit sie unter dem BASIC.SYSTEM noch zulässig sind, z.B. zu den Problemen Hires-Grafik und FRE-Befehl, INPUT bei Strings mit Kommas und Doppelpunkten, Speichern von Zahlen-Arrays als Binärfiles, Simulation des fehlenden MON-Befehls usw. Sodann folgt eine umfangreiche Sammlung von überwiegend in Assembler geschriebenen, sofort einsatzfähigen Hilfsprogrammen:
– Dateileseprogramme zum Einlesen von Dateien beliebiger Größe zwecks Ausgabe als ASCII- oder Hexdump auf dem Bildschirm bzw. Drucker.
– Diskettenkopierprogramme für 1-Drive- und 2-Drive-Besitzer (analog zu COPYA).
– Dateikopierprogramme analog zu FID. Das eine der beiden Programme ist fast ausschließlich in Applesoft geschrieben, um die Leistungsfähigkeit der BASIC.SYSTEM-Befehle zu veranschaulichen.
– Konvertierungsprogramm für die Richtung DOS 3.3 nach ProDOS.
– Ferner Diskettenvergleichsprogramm, Formatierungsprogramm (versionsunabhängig!), Bad-Block-Routine u.a.

Die ProDOS-Analyse
von A. Schäpers, Juli 1985, ca. 450 Seiten, ca. DM 54,-

Auch wenn Arne Schäpers jetzt rot wird, muß es einmal an dieser Stelle gesagt werden: Sein Buch ist die umfangreichste und detaillierteste Darstellung, die jemals ein Apple-Betriebssystem erfahren hat. In der nunmehr zur Neige gehenden DOS-3.3-Ära war „Beneath Apple DOS" von D. Worth und P. Lechner die „Bibel" für alle, die wissen wollten, was DOS 3.3 zum „Ticken" bringt. Den Nachfolgeband „Beneath Apple ProDOS", der im Sommer 1984 erschien, empfand ich persönlich als eine herbe Enttäuschung, weil dort im wesentlichen nur das repetiert wurde, was bereits im „ProDOS Technical Reference Manual" stand. Wer deshalb die teilweise „verqueren Innereien" von ProDOS bis zum letzten Byte – um nicht zu sagen Bit – kennenlernen möchte, der sollte sich diese voluminöse Werk von Arne Schäpers zulegen, welches das komplette Betriebssystem (Urlader, MLI, Disk-Driver, RAM-Disk-Driver und Thunderclock-Routine) – mit Ausnahme des BASIC.SYSTEMs, das zu einem späteren Zeitpunkt gesondert behandelt wird – mit umfangreichen Kommentaren und Übersichtstabellen disassembliert. Während der zweite Band von „ProDOS für Aufsteiger" fertige Utilities enthält, die man auch dann anwenden kann, wenn man über keine Assemblerkenntnisse verfügt, wendet sich die „ProDOS-Analyse" an den reinen Assemblerprogrammierer, dem sich jedoch gerade wegen der minutiösen theoretischen Sezierung von ProDOS völlig neue programmierpraktische Perspektiven eröffnen. Durch die parallele Darstellung aller bisher vorliegenden Versionen – insbesondere 1.0.2 und 1.1.1 – ergeben sich zudem interessante Einsichten in den „Programmierstil" der Firma Apple. Ferner können Programmierer, die ProDOS versionsabhängig patchen möchten, hier genau überblicken, wo gepatcht werden muß, damit dies keine negativen Konsequenzen hat.

Daß auch das BASIC.SYSTEM von uns genauer unter die Lupe genommen werden muß und wird, zeigt das nachfolgende Beispielprogramm. Wenn man es – nachdem man es unbedingt zuvor auf Diskette gespeichert hat! – dreimal startet, und zwar mit

RUN
RUN 130
RUN 140

stellt man beim dritten Durchlauf fest, daß es sich selbst zerstört hat. Dieser Bug wurde in einem Leserbrief an die amerikanische „Nibble" erwähnt, doch wurde dort keine Lösung des Problems angeboten und im übrigen gemutmaßt, daß der Bug nur dann auftritt, wenn das Applesoft-Programm weniger als 1 Block (= 512 Bytes) lang ist. Dies ist jedoch nicht der Fall. Vielmehr tritt die Programmzerstörung exakt dann auf, wenn *unmittelbar* nach dem LOAD oder RUN eines Applesoft-Programms von Diskette eine sequentielle Textdatei mit OPEN – READ Feld-Nummer größer 0 eröffnet wird. Danach wird dieses Feld und nur dieses ab $0801 (Programmanfang) im Speicher abgelegt, und zwar als Zeichenkette plus Return mit Bit 7 off (gilt auch für Zahlen). Daß man in Betriebssystemen immer wieder den einen oder anderen Bug findet, ist nichts Neues, daß jedoch Betriebssysteme auch Programme zerstören, war mir bislang unbekannt. Ab der BASIC.SYSTEM Version 1.1, die zu PRODOS 1.1.1 vom Herbst 1984 gehört, ist dieser Bug beseitigt. Doch wer bekommt schon diese Version? Als ich in *diesem* Jahr für die Peeker-Redaktion einen Apple IIc erwarb, befand sich auf der Systemdiskette („System-Dienstprogramme") noch die Version 1.0.2, und die Diskette „Einführung in BASIC" enthält sogar noch die Version PRODOS 1.0! Sollte also die gekaufte Version nur dazu gedacht sein, daß man angeregt wird, zusätzlich die (hoffentlich dann weitgehend) fehlerfreie Version zu kaufen?

```
100 REM FIELD.BUG:F > 0
110 POKE 768,2
120 PRINT CHR$ (4) "OPEN
    XXX": PRINT CHR$ (4)
    "WRITE XXX": FOR X = 1 TO
    10: PRINT X: NEXT : PRINT
    CHR$ (4) "CLOSE": GOTO
    150
130 PRINT CHR$ (4) "OPEN
    XXX": PRINT CHR$ (4)
    "READ XXX": PRINT CHR$
    (4) "CLOSE": REM Trick-
    Zeile
140 PRINT CHR$ (4) "OPEN
    XXX": PRINT CHR$ (4)
    "READ XXX,F"1: INPUT X:
    PRINT CHR$ (4) "CLOSE"
150 LIST
160 IF PEEK (768) = 2 THEN
    POKE 768,1: PRINT : PRINT
    "Jetzt RUN 130 tippen":
    PRINT CHR$ (4) "LOAD
    FIELD.BUG"
170 IF PEEK (768) = 1 THEN
    POKE 768,0: PRINT : PRINT
    "Jetzt RUN 140 tippen":
    PRINT CHR$ (4) "LOAD
    FIELD.BUG"
180 REM Ohne Trick-Zeile wird
    Applesoft-Programm zerstört!
190 REM Gilt für ProDOS-BASIC-
    .SYSTEM bis Version 1.0.2
```

peeker Börse

Gelegenheitsanzeigen

Sie können unter dieser Rubrik zu einem besonders günstigen Preis

- Ihre Hardware und Software verkaufen
- Ihre Hard- und Software suchen
- Kontakte knüpfen und vieles mehr

Musteranzeige privat

1 Druckzeile à 32 Buchstaben nur DM 5,– zuzügl. ges. MwSt.

Beispiel:

Verkaufe neuwertigen Typenraddrucker mit Apple-Interface. Preis auf Anfrage. Tel. 007

nur DM 17,10 inkl. MwSt.

Musteranzeige gewerblich

Die einspaltige Millimeterzeile (42 mm breit) kostet 5,50 zuzügl. ges. MwSt.

Beispiel:

Neu im Angebot: Professionelle, separate Tastatur für Apple II plus 16 Funktionstasten und separatem Ziffernblock. Fa. Keyboard & Co.

nur DM 137,94 inkl. MwSt.

peeker MAGAZIN FÜR APPLE-COMPUTER

peeker-Börse

AUFTRAG FÜR KLEINANZEIGEN

Bitte veröffentlichen Sie in der nächsterreichbaren Ausgabe nachstehenden Text unter folgender Rubrik:

☐ suche Hardware ☐ suche Software ☐ Tausch ☐ Verschiedenes
☐ biete Hardware ☐ biete Software ☐ Kontakte ☐ Chiffre

Bitte jeweils 32 Buchstaben pro Zeile–einschließlich Satzzeichen und Wortzwischenräume. Bitte Absender nicht vergessen.
Chiffregebühr DM 6,– zuzügl. MwSt.

peeker MAGAZIN FÜR APPLE-COMPUTER

Produkt-Karte

Zu der in »**peeker**«, Heft_____, Seite_____ erschienenen
☐ Anzeige ☐ Produktbesprechung

über _____

bitte ich um detaillierte Information.

Ich wünsche ☐ Prospekt, Datenblatt ☐ Preisliste ☐ schriftliches Angebot ☐ tel. Rückruf

Menge	Produkt und Bestellnummer	à DM	gesamt DM

gebe ich nebenstehende Bestellung unter Anerkennung Ihrer in der Anzeige genannten Liefer- und Zahlungsbedingungen auf.

Unterschrift (für Jugendliche unter 18 Jahren der Erziehungsberechtigte)

peeker MAGAZIN FÜR APPLE-COMPUTER

Info-Karte

Schreiben Sie uns, wenn Sie Fragen, Ergänzungen oder Anregungen zu Aufsätzen in »**peeker**« haben:

peeker-Börse

Vorname, Name

Beruf

Straße

Wohnort

PLZ

Bitte veröffentlichen Sie den umstehenden Text von _____ Zeilen à _____ DM in der nächsterreichbaren Ausgabe von **»peeker«**

Bei Angeboten: Ich bestätige, daß ich alle Rechte an den angebotenen Sachen besitze

Datum Unterschrift

Produkt-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

Anschrift der Firma angeben, bei der Sie bestellen bzw. von der Sie Informationen wünschen

Info-Karte

Karte bitte vollständig ausfüllen

Vorname, Name

Firma

Straße

PLZ/Ort

Telefon mit Vorwahl

ANTWORTKARTE

peeker-Börse
Anzeigen-Service

Dr. Alfred Hüthig Verlag

Postfach 10 28 69

6900 Heidelberg 1

POSTKARTE

Firma

Straße

PLZ/Ort

POSTKARTE

peeker
Redaktion

Postfach 10 28 69

6900 Heidelberg 1

Produkt-Karte

Wünschen Sie weitere Informationen zu einem der im Heft vorgestellten Produkte?

Nichts einfacher als das.
Produkt-Karte ausfüllen, mit 60-Pfennig frankieren und absenden.

Vorher aber nicht vergessen:
kreuzen Sie an, welchen Informationswunsch Sie haben.

Damit erleichtern Sie dem Hersteller eine gezielte Beantwortung Ihrer Anfrage.

Zum Schluß tragen Sie auf der Rückseite die genaue Anschrift des Inserenten/Herstellers und Ihre vollständige Firmenanschrift ein.

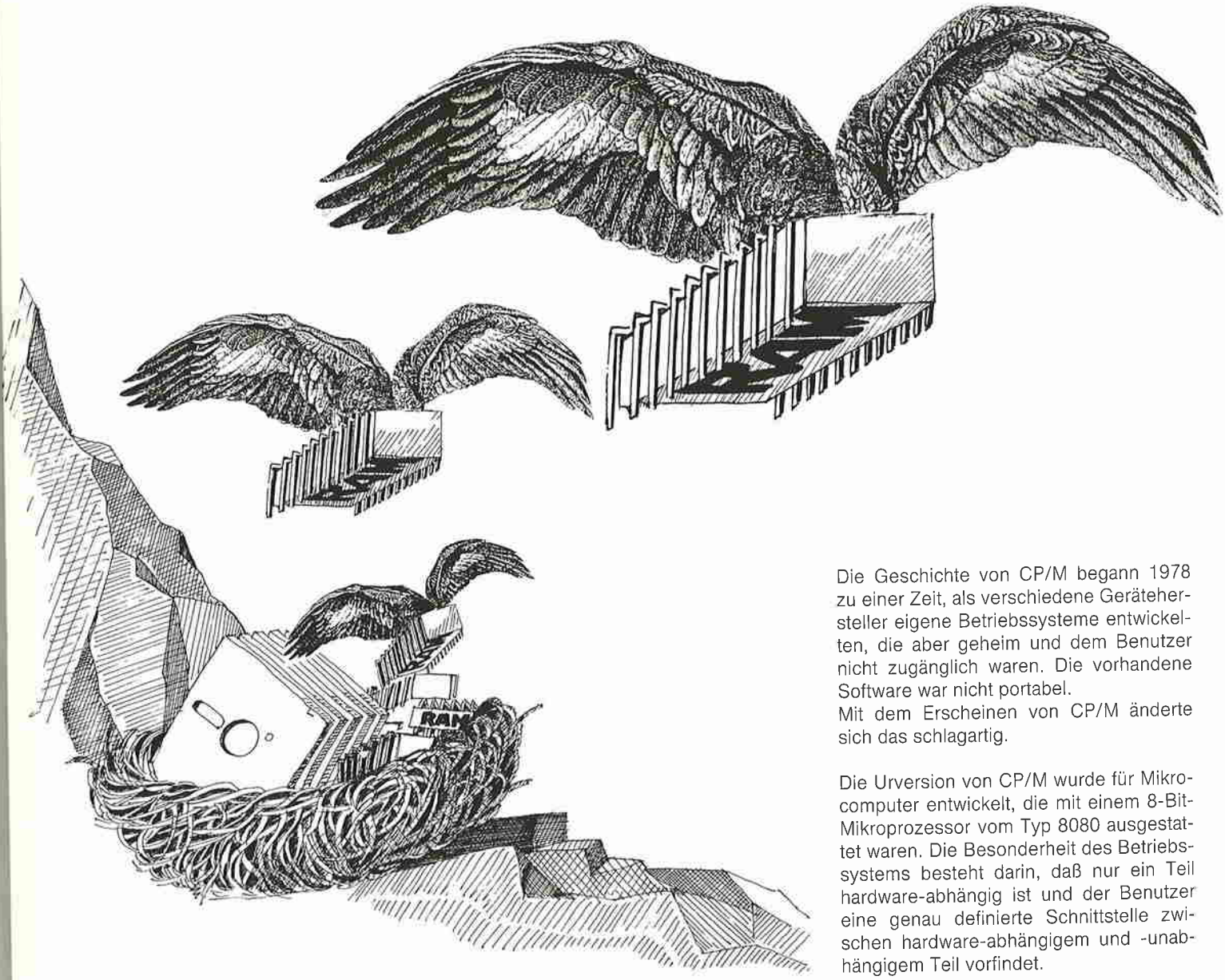
PEEKER
MAGAZIN FÜR APPLE-COMPUT

Apple-CP/M

Mit einem RAM-Disk-Driver für die 64K-Karte

von Karl-Walter Bott

Das Betriebssystem CP/M (Control Program for Microcomputer) ist immer noch eines der weitestverbreiteten Betriebssysteme, obwohl mit dem Einstieg in die 16-Bit-Mikroprozessortechnik eine Reihe von neuen Betriebssystemen wie UNIX oder MS-DOS entwickelt worden sind, die wesentlich leistungsfähiger sind als das auf den Apple-II-Computern verfügbare CP/M. Die Hauptgründe hierfür sind ein großes Software-Angebot und der relativ geringe Preis für ein professionelles Betriebssystem.



Die Geschichte von CP/M begann 1978 zu einer Zeit, als verschiedene Gerätehersteller eigene Betriebssysteme entwickelten, die aber geheim und dem Benutzer nicht zugänglich waren. Die vorhandene Software war nicht portabel.
Mit dem Erscheinen von CP/M änderte sich das schlagartig.

Die Urversion von CP/M wurde für Mikrocomputer entwickelt, die mit einem 8-Bit-Mikroprozessor vom Typ 8080 ausgestattet waren. Die Besonderheit des Betriebssystems besteht darin, daß nur ein Teil hardware-abhängig ist und der Benutzer eine genau definierte Schnittstelle zwischen hardware-abhängigem und -unabhängigem Teil vorfindet.

1. Aufbau des Betriebssystems CP/M

CP/M besteht im wesentlichen aus fünf Modulen:

a) Page 0
Die Page 0 (0000H-00FFH) ist ausschließlich für CP/M reserviert und enthält wichtige Code- und Datensegmente. (Hinweis: Hexadezimale Adressen werden im folgenden für CP/M bzw. Z80 mit nachgestelltem „H" und für 6502 mit vorangestelltem „$" gekennzeichnet.)

b) TPA (Transient Program Area)
Sie beginnt in der Seite 1 ab 0100H. Alle Anwenderprogramme werden ab dieser Adresse von Diskette in den Hauptspeicher geladen und an der Adresse 0100H gestartet.

c) CCP (Console Command Processor)
Dies ist die eigentliche Schnittstelle zum Benutzer. Hier werden alle über die Tastatur eingegebenen Zeichen und Befehle verarbeitet und interpretiert. Die Adresse, ab der der CCP in den Speicher gebracht wird, hängt von der Größe des verfügbaren Gesamtspeicherraums ab.

d) BDOS (Basic Disk Operating System)
Das BDOS verwaltet alle angeschlossenen Diskettenlaufwerke und ist für sämtliche Ein- und Ausgabe-Operationen zuständig. Dem Benutzer wird hier eine Reihe von Grundfunktionen zur Verfügung gestellt, z.B. Laufwerk auswählen, Datei öffnen oder Zeichen von der Konsole lesen. Das BDOS ist ebenfalls völlig von der Hardware unabhängig und bildet mit dem CCP die Basis des Betriebssystems. Das BDOS wird ab Adresse CCP + 0800H in den Hauptspeicher geladen.

e) BIOS (Basic Input Output System)
Das BIOS ist der einzige hardware-abhängige Teil des Betriebssystems und führt alle Ein- und Ausgabe-Funktionen aus, z.B. Daten auf Drucker, Terminal oder Diskette bringen. Das BIOS wird ab Adresse CCP + 1600H in den Speicher geladen.

2. Apple II und CP/M

Die Computer der Apple-II-Serie sind in ihrer Grundausstattung nicht CP/M-fähig: Der Apple besitzt keinen Mikroprozessor vom Typ 8080 oder Z80. Weiterhin wird kein RAM-Bereich ab Adresse 0100H zur Verfügung gestellt, der durchgehend dem Benutzer zur Verfügung steht, da z.B. ab Adresse $0400-$07FF der Bildschirmspeicher angesprochen wird und der Bereich ab $C000-$CFFF ebenfalls tabu ist, da über diese Adressen die wichtigen Softswitches aktiviert werden und Adreßbereiche für Speicher auf Zusatzkarten in den Slots 1-7 reserviert sind.
Abhilfe schafft hier die Z80-Karte. Gemeint ist damit die „normale" Z80-Karte, die keinen Zusatzspeicher und keinen eigenen Systemtakt besitzt. Sie ermöglicht den Betrieb von CP/M 2.2; CP/M 3.0 ist nicht ablauffähig. Sie erfüllt im wesentlichen zwei Aufgaben.
Zum einen wird eine Umkodierung der Adressen vorgenommen, so daß ein durchgehender RAM-Bereich ab Adresse 0000H-DFFFH zur Verfügung steht (**Bild 1**). Der wichtige Softswitchbereich ab $C000 kann unter CP/M mit den Adressen ab E000H angesprochen werden. Das benötigte RAM stellt die Language-Card Bank 2 bereit. Zum anderen wird der benötigte Mikroprozessor zur Verfügung gestellt.

Bild 1

```
0000H-0FFFH  =  1000-$1FFF
1000H-1FFFH  =  2000-$2FFF
2000H-2FFFH  =  3000-$3FFF
3000H-3FFFH  =  4000-$4FFF
4000H-4FFFH  =  5000-$5FFF
5000H-5FFFH  =  6000-$6FFF
7000H-7FFFH  =  8000-$8FFF
8000H-8FFFH  =  9000-$9FFF
9000H-9FFFH  =  A000-$AFFF
A000H-AFFFH  =  B000-$BFFF
B000H-BFFFH  =  D000-$DFFF: Bank 2 der LC
C000H-CFFFH  =  E000-$EFFF: Bank 2 der LC
D000H-DFFFH  =  F000-$FFFF: Bank 2 der LC
E000H-EFFFH  =  C000-$CFFF
F000H-FFFFH  =  0000-$0FFF: Zeropage, Stack, Bildschirm
```

Bild 2

Nr.	Name	Funktion
$00	BOOT	Kaltstart
$01	WBOOT	Warmstart
$02	CONST	Konsolstatus
$03	CONIN	Konsoleingabe
$04	CONOUT	Konsolausgabe
$05	LIST	Druckerausgabe
$06	PUNCH	Stanzerausgabe
$07	READER	Leserausgabe
$08	HOME	Diskettenlesekopf in 0,0-Position
$09	SELDSK	Auswahl des Laufwerks
$0A	SETTRK	Auswahl der Spur
$0B	SETSEC	Auswahl des Sektors
$0C	SETDMA	DMA-Adresse setzen
$0D	READ	Lesen eines Track/Sectors
$0E	WRITE	Schreiben eines Track/Sectors
$0F	LISTST	Druckerstatus
$10	SECTRAN	Übersetzung logisch./physik.Sektor

Quelle: c't 1984, Heft 9, S. 82

Eine Besonderheit des Apple-CP/M-Systems ist die Zusammenarbeit des Z80 mit dem 6502, der als Slave-Prozessor dient, da über ihn alle Ein- und Ausgabe-Operationen abgewickelt werden, die auf der Maschinenebene ablaufen.

3. Die Zusammenarbeit zwischen Z80 und 6502 unter CP/M

Bild 2 zeigt das Programm, welches die eigentliche Schnittstelle zwischen Z80 und 6502 darstellt und von der CP/M-Diskette bei einem Kaltstart ab Adresse $03C0 (F3C0H) in den Speicher gebracht wird. Wichtig ist noch die Speicherstelle $03DE. Sie enthält das höherwertige Byte der Slotadresse, in der die Z80-Karte installiert ist z.B. E4H für Slot 4. Diese Speicherstelle und der entsprechende Operand des STA-Befehls ab Adresse $03C6 werden bei jedem Kaltstart initialisiert. Daraus ergibt sich folgende prinzipielle Vorgehensweise bei der Aktivierung des 6502-Prozessors, wenn der Z80 aktiv ist:

Zunächst muß die Adresse festgelegt werden, an dem der 6502 sein Unterprogramm ausführen soll. Dies geschieht dadurch, daß die gewünschte Adresse in die Speicherstellen $03D0, $03D1 geschrieben wird. Von Bedeutung ist auch hier die beschriebene Umkodierung der Adressen.

Angenommen, wir wollen eine 6502-Routine aus dem Apple-ROM aufrufen, z.B. PRERR mit der Adresse $FF2D, so können wir das mit folgender Z80-Befehlssequenz erreichen:

```
LD HL,0FF2DH
LD (0F3D0),HL
LD HL,(0F3DEH)
LD (HL),A
```

Der erste Befehl lädt das HL-Register mit der Adresse des 6502-Unterprogramms, der zweite schreibt die 16-Bit-Adresse in die Speicherstellen $03D0, $03D1. Von Bedeutung ist hier, daß die Z80-Adresse F3D0H als Zieladresse angegeben wird, da sie der Adresse $03D0, $03D1 entspricht, wenn der 6502 aktiv ist (Bild 1).
Der dritte Befehl holt die Adresse des Slots, in dem die Z80-Karte installiert ist. Auch hier ist die Umkodierung der Adressen zu berücksichtigen.
Der Befehl lädt den Z80-Akkumulator in die Slotadresse. Es kommt hierbei nur darauf an, daß die Slotadresse auf dem Adreßbus des Z80 anliegt und in die angegebene Adresse geschrieben wird. Welche Daten anliegen, ist unerheblich analog zu dem Betätigen eines Softswitches unter Kontrolle des 6502 (z.B. STA $C000). Der Z80 wird dadurch angehalten und der Adreß- und Datenbus für den 6502 freigegeben. Dessen Programmzähler wird mit der Adresse $03C9 geladen, so daß er seine Arbeit an der Adresse $03C9 aufnimmt.
An dieser Stelle werden in bekannter Weise zunächst das Apple-ROM lesefähig gemacht und die CPU-Register geladen. Mit dem Befehl JSR $xxxx ab Adresse $03CF wird das eigentliche 6502-Unterprogramm aufgerufen. Die Adresse $xxxx ist vorher vom Z80 definiert worden. Nach der Rückkehr wird das ROM auf jeden Fall wieder lesefähig gemacht und die 6502-Register werden gerettet.
Mit dem Sprung nach $03C0 wird die Rückkehr zum Z80 eingeleitet: Language-Card Bank 2 schreib- und lesefähig machen und mit „STA $Cn00" den Z80 aktivieren. Der Z80 setzt die Arbeit mit dem nächsten Befehl nach „LD (HL),A" fort. Stackpointer und sämtliche Registerinhalte werden nicht verändert.

SWITCHDEMO.HEX

```
:100100003E003224F0212E117D3245F07C3247F042
:1001100213ADB22D0F32ADEF3773E03CD2201C35E
:10012000000003D5782822A0100160005F19E95A3803
:10013000302D36353032205357495443482D4445ED
:03014000D4F002000000000000000000000000000000
```

SWITCHDEMO

```
                ; *** SWITCHDEMO ***
                ;
0100                            ORG 0100H
                ;
F047 =          YREG            EQU 0F047H
F045 =          AREG            EQU 0F045H
DB3A =          STROUT          EQU 0DB3AH          ;String Output
F024 =          HCURS           EQU 0F024H          ;Horizontaler Cursor
F3D0 =          VEC             EQU 0F3D0H
F3DE =          CPU             EQU 0F3DEH
                ;
0003 =          CONIN           EQU 00003H
0000 =          WBOOT           EQU 00000H
                ;
0100 3E00                       MVI A,0
0102 3224F0                     STA HCURS           ;HTAB = 0
                ;
0105 212E11                     LXI H,STRING+1000H  ;Adresse des Strings
0108 7D                         MOV A,L
0109 3245F0                     STA AREG            ;LSB --> A-Reg. 6502
010C 7C                         MOV A,H
010D 3247F0                     STA YREG            ;MSB --> Y-Reg. 6502
0110 213ADB                     LXI H,STROUT        ;Up-Adresse übergeben
0113 22D0F3                     SHLD VEC
0116 2ADEF3                     LHLD CPU            ;Slotadr. holen
0119 77                         MOV M,A             ;6502 aktivieren
                ;
011A 3E03                       MVI A,CONIN         ;FKT-Nr. für CONIN
011C CD2201                     CALL BIOS           ;durch BIOS ausführen
011F C30000                     JMP WBOOT
                ;
                ; Einsprungadresse der BIOS-Funktion berechnen und
                ; Funktion ausführen
                ;
0122 3D         BIOS            DCR A
0123 57                         MOV D,A
0124 82                         ADD D
0125 82                         ADD D
0126 2A0100                     LHLD WBOOT+1
0129 1600                       MVI D,0
012B 5F                         MOV E,A
012C 19                         DAD D
012D E9                         PCHL
                ;
012E 5A3830 2D36STRING          DB "Z80-6502 SWITCH-DEMO"
0142 00                         DB 00H
0143                            END
```

Wenn wir das Programm **SWITCHDEMO** betrachten, stellen wir fest, daß eine einfache Parameterübergabe zwischen CP/M und 6502 möglich ist. Die Parameter können entweder direkt über die Zero-Page übergeben werden, wie dies durch die Übergabe der horizontalen Cursorposition demonstriert wird, oder durch „Poken" in die 6502-Register-Save-Area, da die 6502-Register bei jeder Aktivierung durch die Schnittstelle 6502-Z80 geladen werden.
Nach dem Start des Programms SWITCHDEMO erscheint auf dem Bildschirm der Text „Z80-6502 Switch-Demo", der durch die Applesoft-Routine STROUT ausgegeben wird. Danach erfolgt ein Rücksprung zu CP/M, von wo über die BIOS-Funktion CONIN auf einen Tastendruck gewartet wird, der einen CP/M-Warmstart auslöst.

4. Apple IIe und CP/M

Auch auf diesem Computer ist CP/M genauso lauffähig wie auf einem Apple II. Unangenehm ist nur, daß nach einem CP/M-Warmstart die Firmware der neuen 80-Zeichenkarte den gesamten Bildschirm löscht, und somit Informationen wie z.B Fehlermeldungen eines Compilers, die für weitere Operationen benötigt werden, über den Bildschirm nicht zu verarbeiten sind. Ebenfalls wird die erweiterte 80-Zeichenkarte (= 64K-Karte) nicht unterstützt, so daß man praktisch 64K Speicherraum „verschenkt".

5. RAM-Disk für Apple IIe mit erweiterter 80-Zeichenkarte

Der hier abgedruckte RAM-Disk-Driver ist für einen Apple IIe mit erweiterter 80-Zeichenkarte mit zwei Diskettenlaufwerken unter Apple-CP/M 56K konfiguriert.
Um die Wirkungsweise der RAM-Disk und deren Wechselwirkung mit den real existierenden Laufwerken zu verdeutlichen, ist es erforderlich, auf die CP/M-Module BIOS und BDOS näher einzugehen. Auch kann dann das Programm an andere CP/M-Versionen angepaßt werden, und der Leser ist in der Lage, sein eigenes Diskettenformat zu implementieren. (Theoretisch läßt sich der RAM-Disk-Driver auch auf einem Apple IIc mit CP/M-Karte implementieren, doch wurde dies nicht ausgetestet.)

GETBIOS

```
                ;   *** GETBIOS ***
                ;
                ;   GETBIOS berechnet die Adresse einer BIOS-Funktion (1...16)
                ;   Die Nummer der Funktion wird im A-Register Übergeben.
                ;   Das HL-Register liefert die Adresse an das aufrufende
                ;   zurück.
                ;
Ø1ØØ                ORG Ø1ØØH
                ;
ØØØØ =          WBOOT   EQU ØØØØH
                ;
Ø1ØØ 3D             DCR  A              ; A = A - 1
Ø1Ø1 57             MOV D,A
Ø1Ø2 82             ADD D               ; A = A * 3
Ø1Ø3 82             ADD D
Ø1Ø4 2AØ1ØØ         LHLD WBOOT+1        ; Warmstart-Vektor holen
Ø1Ø7 16ØØ           MVI D,Ø
Ø1Ø9 5F             MOV E,A             ; Offset in das DE-Reg.
Ø1ØA 19             DAD D               ; Adresse = Offset + Warmboot
Ø1ØB C9             RET                 ; Zurück
Ø1ØC                END
```

GETBIOS.HEX

```
:ØCØ1ØØØØ3D5782822AØ1ØØ16ØØ5F19C9D9
:ØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØØ
```

5.1. Arbeitsweise des BIOS

Dies ist, wie schon erwähnt, der hardwareabhängige Teil des Betriebssystems. Das BIOS kommuniziert mit BDOS oder mit einem Benutzerprogramm durch eine Sprungtabelle, wodurch 17 definierte Funktionen erreicht werden können (Bild 2). Die Adresse, ab der diese Sprungtabelle im Hauptspeicher des Rechners steht, ist von der maximalen Speicherkapazität abhängig. Wenn ein Anwenderprogramm BIOS-Funktionen aufrufen will, muß es die Adresse der Sprungtabelle im Speicher kennen, um über diese die Adresse der selektierten BIOS-Funktion zu ermitteln. Das Programm **GETBIOS** berechnet die Einsprungadresse der gewünschten BIOS-Funktion, indem es die Adresse des ersten Sprungvektors (Funktion WBOOT) ermittelt, die durch den Inhalt der Speicherstellen 2 und 3 der Page 0 gegeben ist. Daraus können dann die Einsprungadressen aller BIOS-Funktionen errechnet werden.
Von den aufgeführten BIOS-Funktionen interessieren uns nur diejenigen, die mit den Diskettenoperationen in Verbindung stehen, insbesondere die Funktionen 9-14 und 16, die im folgenden beschrieben werden.

SELDSK
Diese Funktion aktiviert das vom Benutzer spezifizierte Laufwerk. Als Parameter wird im C-Register die CP/M-Laufwerksnummer übergeben: 0=A, 1=B, ..., 15=P. Konnte die Funktion erfolgreich ausgeführt werden, wird im HL-Register die Adresse des DPH („Disk Parameter Header"), auf dessen Bedeutung später eingegangen wird, zurückgegeben. Tritt ein Fehler auf, so enthält das HL-Register den Wert 0000H.

SETTRK
Hiermit wird die Nummer der Spur ausgewählt (BC-Register), auf die beim nächsten Diskettenzugriff geschrieben oder von der gelesen werden soll.

SETSEC
Für alle folgenden Zugriffe auf das aktuelle Laufwerk wird die gewünschte Sektornummer (BC-Register) festgelegt.

SETDMA
Hier wird die Adresse eines 128-Byte-Puffers (BC-Register) übergeben. Alle folgenden Leseoperationen von der aktuellen Diskette schreiben in diesen Bereich, alle Schreiboperationen übertragen den Inhalt ab der selektierten Pufferadresse auf das aktuelle Laufwerk.

READ
Lesen eines gewählten Track/Sector von der aktuellen Diskette. Nach einem Lesefehler enthält das A-Register den Wert 0, sonst 1.

WRITE
Schreiben eines gewählten Track/Sector aus dem Puffer auf das aktuelle Diskettenlaufwerk. Es werden die gleichen Parameter wie bei READ zurückgegeben.

SECTRAN
Um die Geschwindigkeit des Diskettenzugriffs zu erhöhen, kann ein logischer Sektor in einen physikalischen Sektor übersetzt werden. Standard-CP/M benutzt einen „Interleaving-Faktor" von sechs, d.h. nach zwei logisch aufeinanderfolgenden Sektoren werden sechs physikalische Sektoren übersprungen. Die SECTRAN-Funktion wird bei Apple-CP/M nicht benutzt, was bedeutet, daß keine physikalischen Sektoren übersprungen werden sollen.

Dem BIOS wird noch ein Satz von Tabellen vorangestellt, die wichtige Parameter für die angeschlossenen Diskettenlaufwerke beinhalten. Für jedes Diskettenlaufwerk wird ein **DPH** (Disk Parameter Header) angelegt, der 16 Bytes einnimmt. Der schematische Aufbau eines DPH ist in **Bild 3** dargestellt.
Die ersten 2 Bytes geben die Adresse eines Vektors für die Übersetzung von logischen in physikalische Sektoren für das Laufwerk an. Ist die Adresse 0000H, so findet keine Übersetzung statt (Apple-CP/M).
Die nächsten 6 Bytes sind für das BDOS reserviert und dienen als Scratchspace.
Die 2 folgenden Bytes geben eine Adresse an, die einen Arbeitsbereich von 128 Bytes definiert, den das BDOS für Directory-Operationen benötigt. Es wird für jedes Laufwerk der gleiche Directory-Puffer verwendet.
Die 2 folgenden Bytes stellen ebenfalls einen Zeiger dar, der auf den sog. **DPB** (Disk Parameter Block) zeigt. Dieser Block enthält die eigentliche Diskettenbeschreibung. Laufwerke mit gleichen Eigenschaften sollten denselben DPB benutzen.

Byte 12 und 13 geben die Adresse der Prüfsummentabelle an. Die letzte 16-Bit-Zahl liefert die Adresse des Diskettenbelegungsplans im Speicher, der ebenfalls vom BDOS verwaltet wird.
Apple-CP/M 56K legt 6 DPHs ab Adresse DA33H und 1 DPB ab Adresse DA93H an. Um den Aufbau des DPBs zuverstehen, betrachte man **Bild 4**. Hier ist der Aufbau eines DPBs dargestellt.
SPT gibt die Anzahl der Spuren auf der Diskette an. BSH und BLM bestimmen zusammen die Blockgröße auf der Diskette, wobei sich BSH berechnet zu log2 (Blockgröße/128). Die Blockgröße ist ein Vielfaches von 128, bei den Apple-Disketten 1024 Bytes. BLM ist die sogenannte „Blockmask" die sich aus 2 ↑ (BSH-1) ergibt. EXM ist die Extensionsmaske und hängt von der Blockgröße ab. Der Wert von EXM kann aus **Bild 6** bestimmt werden. **DSM** bestimmt die totale Speicherkapazität der Diskette, wobei (DSM+1) · Blockgröße die gesamte Kapazität der Diskette ohne Systemspuren ergibt. DRM legt die Anzahl der möglichen Directory-Einträge fest. Hier muß die maximale Anzahl -1 eingetragen werden. CKS gibt die Größe des Directory-Prüfvektors an: CKS = (DRM+1) / 4.
Auch das Directory selbst belegt Speicherplatz auf der Diskette. AL0 und AL1 geben die dafür reservierten Blöcke auf der Diskette an. Jeder Eintrag benötigt 32 Bytes. Bei 48 möglichen Directory-Einträgen auf der Diskette ergeben sich 48 · 32 = 1536 Bytes, was bei einer Blockgröße von 1024 Bytes zwei Blöcke in Anspruch nimmt. Diese Blöcke werden durch Setzen eines Bits in AL0 und AL1 markiert, und zwar beginnend mit dem höherwertigen Bit von AL0. Somit erhalten wir bei den angenommenen Größen für AL0 $C0 und für AL1 $00.
Die letzten 2 Bytes, die mit OFF bezeichnet sind, enthalten die Anzahl der Systemspuren. Apple-CP/M reserviert hier drei (0..2).

XLT	SCRATCH	DIRBUF	DPB	CSV	ALV
2 Bytes	6 Bytes	2 Bytes	2 Bytes	2 Bytes	2 Bytes

Bild 3

SPT	BSH	BLM	EXM	DSM	DRM	AL0	AL1	CKS	OFF

Bild 4

Sek. \ Spur	0	1	2	10	11	12	13	14	15
0	0800	1800	2800	A800	D000	D000	E000	F000	D000
1	⋮			⋮	⋮	⋮	⋮	⋮	⋮
2	⋮			⋮	⋮	⋮	⋮	⋮	⋮
3	⋮			⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮			⋮	⋮	⋮	⋮	⋮	⋮
14	1600			B600	⋮	⋮	⋮	FEFF	⋮
15	17FF			B7FF	DFFF	DFFF	EFFF	0300	DFFF
					Bank 2 oben	Bank 1 oben			Bank 1 unten

Bild 5

Blockgröße	DSM <255	DSM >255
1024	0	–
2048	1	0
4096	3	1
8192	7	3
16384	15	7

Bild 6

5.2. CP/M-Patch

Um eine RAM-Disk zu installieren, muß zunächst eine geeignete Stelle gefunden werden, um die RWTS-Routine zu linken. Soll ein bestimmter Track/Sector durch das BIOS gelesen oder geschrieben werden, wird zu einer 6502-Routine ab $0E03 verzweigt. Dort kann dann ein Sprung zur eigenen RWTS erfolgen. Bevor jedoch die Routine ausgeführt werden kann, muß das BIOS mehrere erforderliche Parameter berechnen.

Die gewählte Laufwerksnummer wird in $03E4, der angesprochene Slot, in dem der zugeordnete Disk-Controller installiert ist, wird mit $10 multipliziert in $03E6 abgelegt. Die Adresse $03EB bestimmt, ob gelesen oder geschrieben werden soll, wobei 1 Schreiben und 2 Lesen bedeutet. Es wird immer auf 2 CP/M-Sektoren (2 * 128 Bytes) auf einmal zugegriffen. Daraus folgt, daß der DMA-Puffer von 128 Bytes nicht sofort auf Diskette geschrieben wird, sondern erst der 256 Bytes große Diskettensektor in einen weiteren Puffer ab $0800 gelesen wird. Erst danach kopiert CP/M den Inhalt des DMA-Puffers entweder in die erste oder zweite Hälfte des Zwischenpuffers. Wenn auf einen anderen Diskettensektor zugegriffen werden soll, wird der Zwischenpuffer bedingungslos auf Diskette gebracht.
Die Spurnummer wird in $03E0, die Nummer des Diskettensektors in $03E1 übergeben.
Von Bedeutung sind das Errorflag ($03EA), das angibt, ob die Diskettenoperation fehlerfrei abgelaufen ist (0 = kein Fehler), und die Speicherstelle DEAEH, welche die Nummer des momentan unter CP/M aktiven Laufwerks angibt (A=0, B=1, C=2...).

5.3. CP/M-RAM-Disk-Driver

Der RAM-Disk-Driver besteht aus zwei Teilen. Der erste Teil befindet sich im AUXZP-Bereich auf der erweiterten 80-Zeichenkarte ab Adresse $FF00 und belegt etwa eine Speicherseite. Die imaginären Spuren der RAM-Disk (**Bild 5**) mit den Nummern 0-10 werden im Zusatzspeicher im AUXMAIN-47,5K-Bereich ab Adresse $0800-$B7FF abgebildet. Die Spuren 11 und 12 entsprechen den beiden Banks der Language-Card ($D000-$DFFF) im AUXZP-Bereich des Zusatzspeichers. Spur 13 und 14 korrespondieren mit mit der gemeinsamen Bank der Language-Card im Zusatzspeicher ($E000-$FEFF). Spur 14, Diskettensektor 15 wird durch die Seite drei ($0300-$03FF) des AUXMAIN-47,5K-Bereichs repräsentiert. Der eigentliche Driver kann vom AUXZP-Bereich des Zusatzspeichers Daten aus den imaginären Spuren 0..14 sowohl in den Zwischenpuffer ab Adresse $0800 auf der Hauptplatine schreiben als auch vom Zwischenpuffer lesen und in den Zusatzspeicher bringen, da sich der Driver in jedem Fall selber lesen kann.

Für die Spur 15 bleibt nur noch die Bank 1 der Language-Card auf der Hauptplatine im Bereich $D000-$DFFF übrig, da nur Bank 2 von CP/M selbst benutzt wird. Um von dieser Spur in den Zwischenpuffer zu lesen, muß sich ein weiterer Driver im „ungebankten" Bereich des Hauptspeichers befinden. Der zweite Driver liegt ab FC70H (= $0C70) in einem von CP/M nicht benötigten Bereich.

5.4. Installation des RAM-Disk-Drivers

Zunächst sind das Applesoft-Programm **RAMDISK.INIT.DOS** und die drei 6502-Programme **AUXDRIVER**, **MOVEDRIVER** und **RAMDISK.FORMATTER** zu erstellen.
MOVEDRIVER verschiebt lediglich den Objekt-Code, der ab Adresse $FF00 lauffähig sein muß, vom Hauptspeicher in den Zusatzspeicher.
Der RAMDISK.FORMATTER initialisiert den gesamten Speicherbereich, den die RAM-Disk verwendet, mit dem Wert $E5.
Als viertes Modul muß noch das Programm **RDSKINIT.COM** auf der CP/M-Diskette vorhanden sein. Es ist in 8080-Assembler geschrieben, da kein Z80-Assembler zur Verfügung stand.
RDSKINIT.COM patcht das CP/M an der Stelle $0E03. Hier wird ein Sprung zur eigenen RWTS-Routine installiert. Des weiteren muß die Anzahl der angeschlossenen Diskettenlaufwerke erhöht werden, indem in den Speicherplatz $03B8 eine 3 eingetragen wird, so daß die RAM-Disk unter Laufwerk „C" angesprochen werden kann. Außerdem wird ein neuer DPB für die RAM-Disk initialisiert und die Adresse in den DPH eingetragen. Bleibt nur noch das „Poken" des Drivers im Hauptspeicher ab Adresse $0C70. Der Driver prüft zuerst, ob das angeschlossene Laufwerk die RAM-Disk ist. Wenn ja, wird der ALTZPAUX-Bereich selektiert und die Language-Card schreib- und lesefähig gemacht und zum Driver im Zusatzspeicher gesprungen. Die Rückkehr erfolgt entweder über RTSDRIVER, wenn auf die Spuren 0...14 zugegriffen wurde, oder durch einen Sprung nach DRIVER2 vom Driver im Zusatzspeicher (Zugriff auf Spur 15). Alle Transferparameter, die ACCESS 2 benötigt, sind bereits errechnet, müssen jedoch vorher von der AUX-Zero-Page in die MAIN-Zero-Page kopiert werden. Am Ende wird das RWTS-Errorflag auf 0 gesetzt, so daß kein BDOS-Error auftreten kann. Der Return zum Z80 wird über die oben beschriebene Schnittstelle ausgeführt.
Wurde ein physisch existierendes Diskettenlaufwerk angesprochen, wird die normale RWTS-Routine ab Adresse $0E10 aufgerufen und zum Z80 zurückgekehrt.

5.5 Starten des RAM-Disk-Drivers

Der RAM-Disk-Driver wird von DOS 3.3 aus gestartet. Booten Sie DOS und starten das Programm RAMDISK.INIT.DOS. Danach können Sie wie gewohnt CP/M boo-

RDSKINIT

```
                  ; *** RDSKINIT ***
                  ; Installiert CP/M-Patch für RAM-Disk
                  ;
Ø1ØØ                        ORG Ø1ØØH
FC7Ø =            PATCHADR  EQU ØFC7ØH      ;Hier CP/M-Patch installieren
F3B8 =            DISKNR    EQU ØF3B8H      ;Anzahl der installierten Laufwerke
FEØ3 =            RWTSPATCH EQU ØFEØ3H
                  ;
DA83 =            DPH       EQU ØDA83H
DA5D =            DPHPATCH  EQU ØDA5DH
                  ;
                  ; Ändern des DPB-Zeigers im DPH für Disk C
                  ;
Ø1ØØ 2183DA                 LXI H,DPH
Ø1Ø3 225DDA                 SHLD DPHPATCH
                  ;
                  ; Neuen DPB für RAM-Disk "C" initialisieren
                  ;
Ø1Ø6 1183DA                 LXI D,DPH
Ø1Ø9 Ø19EØ1                 LXI B,DPB
Ø1ØC 26ØF                   MVI H,15
Ø1ØE 7C           TRANSFER  MOV A,H
Ø1ØF B7                     ORA A
Ø11Ø CA1BØ1                 JZ FERTIG
Ø113 25                     DCR H
Ø114 ØA                     LDAX B
Ø115 12                     STAX D
Ø116 Ø3                     INX B
Ø117 13                     INX D
Ø118 C3ØEØ1                 JMP TRANSFER
                  ;
Ø11B 11Ø3FE       FERTIG    LXI D,RWTSPATCH
Ø11E 3E4C                   MVI A,ØØ4CH     ; JMP START
Ø12Ø 12                     STAX D
Ø121 13                     INX D
Ø122 3E7Ø                   MVI A,ØØ7ØH
Ø124 12                     STAX D
Ø125 13                     INX D
Ø126 3EØC                   MVI A,ØØØCH
Ø128 12                     STAX D
                  ;
Ø129 11B8F3                 LXI D,DISKNR
Ø12C 3EØ3                   MVI A,Ø3        ; 3 Laufwerke A,B,C
Ø12E 12                     STAX D
Ø12F 117ØFC                 LXI D,PATCHADR
Ø132 Ø144Ø1                 LXI B,OPCODE
Ø135 2659                   MVI H,ENDOP-OPCODE
Ø137 7C           NEXT      MOV A,H
Ø138 B7                     ORA A
Ø139 CAØØØØ                 JZ  ØØØØ
Ø13C 25                     DCR H
Ø13D ØA                     LDAX B
Ø13E 12                     STAX D
Ø13F Ø3                     INX B
Ø14Ø 13                     INX D
Ø141 C337Ø1                 JMP NEXT
Ø144 AD83CØ       OPCODE    DB ØADH,83H,ØCØH   ;              LDA $CØ83
Ø147 ADAEFE                 DB ØADH,ØAEH,ØFEH  ;              LDA TMPDSK
Ø14A C9Ø2                   DB ØC9H,Ø2H        ;              CMP #Ø2
Ø14C FØØA                   DB ØFØH,ØAH        ;              BEQ RAMDISK
Ø14E Ø8                     DB Ø8H             ;              PHP
Ø14F 78                     DB 78H             ;              SEI
Ø15Ø 2Ø1ØØE                 DB 2ØH,1ØH,ØEH     ;              JSR RWTS
Ø153 AD81CØ                 DB ØADH,81H,ØCØH   ;              LDA $CØ81
Ø156 28                     DB 28H             ;              PLP
Ø157 6Ø                     DB 6ØH             ;              RTS
Ø158 8DØ9CØ                 DB 8DH,Ø9H,ØCØH    ; RAMDISK      STA ALTZPAUX
Ø15B AD83CØ                 DB ØADH,83H,ØCØH   ;              LDA $CØ83
Ø15E AD83CØ                 DB ØADH,83H,ØCØH   ;              LDA $CØ83
Ø161 4CØØFF                 DB 4CH,ØØH,ØFFH    ;              JMP DRIVER
Ø164 8DØ8CØ                 DB 8DH,Ø8H,ØCØH    ; RTSDRIVER    STA ALTZPMAIN
Ø167 AD81CØ                 DB ØADH,81H,ØCØH   ;              LDA $CØ81
Ø16A 6Ø                     DB 6ØH             ;              RTS
Ø16B A5CE                   DB ØA5H,ØCEH       ; DRIVER2      LDA IND1
Ø16D A6CF                   DB ØA6H,ØCFH       ;              LDX IND1+1
Ø16F A4Ø6                   DB ØA4H,Ø6         ;              LDY IND2
Ø171 8CC8ØC                 DB Ø8CH,ØC8H,ØØCH  ;              STY TMP
Ø174 A4Ø7                   DB ØA4H,Ø7         ;              LDY IND2+1
Ø176 8DØ8CØ                 DB Ø8DH,Ø8,ØCØH    ;              STA ALTZPMAIN
Ø179 85CE                   DB Ø85H,ØCEH       ;              STA IND1
Ø17B 86CF                   DB Ø86H,ØCFH       ;              STX IND1+1
Ø17D 84Ø7                   DB Ø84H,Ø7         ;              STY IND2+1
Ø17F ADC8ØC                 DB ØADH,ØC8H,ØCH   ;              LDA TMP
Ø182 85Ø6                   DB Ø85H,Ø6         ;              STA IND2
Ø184 AD8BCØ                 DB ØADH,8BH,ØCØH   ;              LDA $CØ8B
```

ten und RDSKINIT.COM starten, wenn Sie die RAM-DISK aktivieren wollen.
Der Driver im Hauptspeicher wird bei einem CP/M-Warmstart nicht zerstört. Deshalb muß das Programm RDSKINIT.COM nur nach einem Kaltstart wieder gestartet werden. Alle Daten auf der RAM-Disk und der Driver im Zusatzspeicher bleiben erhalten, bis Sie den Rechner ausschalten.

5.6 Anpassen des RAM-Disk-Drivers an andere CP/M-Versionen

Der RAM-Disk-Driver ist für Apple-CP/M Version 2.20 56K mit zwei Diskettenlaufwerken konfiguriert. Wenn mehr als zwei Laufwerke angeschlossen sind, muß der Wert in Speicherstelle F3B8H entsprechend geändert und der Zeiger auf den DPB im der RAM-Disk zugeordneten DPH angepaßt werden.
Der DPB für die RAM-Disk ist so intialisiert, daß 32 Directory-Einträge möglich sind. Daraus ergibt sich eine Netto-Speicherkapazität von 63K (keine Systemspuren). Der Benutzer kann in den DPB andere Parameter eintragen, um das Pseudo-Laufwerk seinen eigenen Bedürfnissen anzupassen.

5.7. Zusammenfassung

Die Peeker-Sammeldiskette enthält folgende RAM-Disk-Programme:
- RAMDISK.INIT.DOS (Basic)
- T.AUXDRIVER (Source)
- AUXDRIVER (6502)
- T.MOVEDRIVER (Source)
- MOVEDRIVER (6502)
- T.RAMDISK.FORMATTER (Source)
- RAMDISK.FORMATTER (6502)

Das Z80-Programm RDSKINIT.COM sowie die Demos SWITCHDEMO und GETBIOS befinden sich nicht auf der Sammeldisk.
Die RAM-Disk läßt sich wie folgt installieren:
1. Booten von DOS 3.3
2. Von Peeker-Sammeldisk mit
RUN RAMDISK.INIT.DOS Return
Driver auf 64K-Karte schieben.
3. Booten von CP/M
4. Mit RDSKINIT Return
Driver anschließen.
5. Danach RAM-Disk mit
C: Return
ansprechen.

```
Ø187 AD8BCØ                DB ØADH,8BH,ØCØH   ;              LDA $CØ8B
Ø18A AØØØ                  DB ØAØH,ØØ         ;ACCESS2      LDY #Ø
Ø18C B1CE                  DB ØB1H,ØCEH       ;NEXT         LDA (IND1),Y
Ø18E 91Ø6                  DB Ø91H,Ø6         ;             STA (IND2),Y
Ø19Ø 88                    DB Ø88H            ;             DEY
Ø191 DØF9                  DB ØDØH,ØF9H       ;             BNE NEXT
Ø193 A9ØØ                  DB ØA9H,ØØ         ;             LDA #Ø
Ø195 8DEAØ3                DB Ø8DH,ØEAH,Ø3H   ;             STA ERFL
Ø198 AD81CØ                DB ØADH,Ø81H,ØCØH  ;             LDA $CØ81
Ø19B 6Ø                    DB Ø6ØH            ;             RTS
Ø19C ØØ                    DB ØØ              ;TMP          DB ØØ
Ø19D FF          ENDOP     DB ØFFH            ;
                 ;
Ø19E 2ØØØ        DPB       DB 2ØH,ØØ          ;Sektoren pro Spur
Ø1AØ Ø3Ø7                  DB Ø3,Ø7           ;BSH/BLM
Ø1A2 ØØ                    DB ØØ              ;EXM
Ø1A3 3FØØ                  DB 3FH,ØØ          ;DSM
Ø1A5 1FØØ                  DB 1FH,ØØ          ;DRM
Ø1A7 8Ø                    DB 8ØH             ;ALØ
Ø1A8 ØØ                    DB ØØ              ;AL1
Ø1A9 ØØØØ                  DB ØØ,ØØ           ;CKS
Ø1AB ØØØØ                  DB ØØ,ØØ           ;OFF
Ø1AD                       END
```

RDSKINIT.HEX

```
:1ØØ1ØØØØ    2183DA225DDA1183DAØ19EØ126ØF7CB7A2
:1ØØ11ØØØ    CA1BØ125ØA12Ø313C3ØEØ111Ø3FE3E4C34
:1ØØ12ØØØ    12133E7Ø12133EØC1211B8F33EØ312115B
:1ØØ13ØØØ    7ØFCØ144Ø126597CB7CAØØØØ25ØA12Ø34D
:1ØØ14ØØØ    13C337Ø1AD83CØADAEFEC9Ø2FØØAØ87813
:1ØØ15ØØØ    2Ø1ØØEAD81CØ286Ø8DØ9CØAD83CØAD8375
:1ØØ16ØØØ    CØ4CØØFF8DØ8CØAD81CØ6ØA5CEA6CFA455
:1ØØ17ØØØ    Ø68CC8ØCA4Ø78DØ8CØ85CE86CF84Ø7AD39
:1ØØ18ØØØ    C8ØC85Ø6AD8BCØAD8BCØAØØØB1CE91Ø66A
:1ØØ19ØØØ    88DØF9A9ØØ8DEAØ3AD81CØ6ØØØFF2ØØØ7E
:ØDØ1AØØØ    Ø3Ø7ØØ3FØØ1FØØ8ØØØØØØØØØØØØ6AØØØØØØ
```

Hinweis zum Erstellen der 8080-Assemblerprogramme unter CP/M

Dieser Hinweis ist für diejenigen Leser gedacht, die nicht mit den abgedruckten 8080-Quelltexten in Verbindung mit dem CP/M-Assembler arbeiten möchten. Die Programme können direkt mit dem DDT (Dynamic Debbuging Tool) als Hexdump eingegeben werden. Starten Sie von Ihrer CP/M-Diskette des DDT (DDT und Return eingeben). Es meldet sich mit
DDT VERS 2.2
–
Danach geben Sie „S0100" ein. Es folgt die Meldung „0100 01". Hierbei bedeutet „0100" die momentan angesprochene Speicherstelle und „01" deren Inhalt.
Jetzt kann der entsprechende Hexdump – jeweils ein Byte, gefolgt von der Return-Taste – eingegeben werden.
Der „S-Modus" wird nach abgeschlossener Eingabe des Hexdumps mit „." beendet. Verlassen Sie den DDT mit Control-C. Nun muß das erstellte Programm noch abgespeichert werden. Dazu tippen Sie „SAVE 1 Name.COM".

RAMDISK.INIT.DOS

```
100 REM RAMDISK.INIT.DOS
110 HOME :D$ =  CHR$ (4)
120 PRINT "CP/M-RAMDISK-DRIVER WIRD INSTALLIERT ..."
130 PRINT D$"BLOAD AUXDRIVER,A$9000"
140 PRINT D$"BRUN MOVEDRIVER,A$9500"
150 PRINT D$"BRUN RAMDISK.FORMATTER,A$9000"
```

AUXDRIVER

```
                 1   ***************************
                 2   *                         *
                 3   * AUXDRIVER               *
                 4   * AUXZP-Bereich     $FF00 *
                 5   *                         *
                 6   ***************************
                 7   *
                 8            ORG  $FF00
                 9   *
                10   BANK2    EQU  $C083
                11   BANK1    EQU  $C08B
                12   RAMRDA   EQU  $C003
                13   RAMRDM   EQU  $C002
                14   RMWRTA   EQU  $C005
                15   RMWRTM   EQU  $C004
                16   *
                17   RW       EQU  $03EB
                18   BUFF     EQU  $0800
                19   ERFL     EQU  $03EA
                20   TRACK    EQU  $03E0
                21   SECTOR   EQU  $03E1
                22   *
                23   IND1     EQU  $CE
                24   IND2     EQU  $06
                25   *
FF00: A9 00     26   START    LDA  #00
FF02: 85 CE     27            STA  IND1
FF04: 85 06     28            STA  IND2
FF06: AD E0 03  29            LDA  TRACK
                30   *
                31   * Spurnummer in AUXZP ablegen
                32   *
FF09: 8D E3 FF  33            STA  STRACK
                34   *
                35   * Adresse im Speicher berechnen
                36   *
FF0C: 20 68 FF  37            JSR  CALCADR
FF0F: AD C0 FF  38            LDA  ADR+1
                39   *
                40   * Kein Fehler aufgetreten
                41   *
FF12: D0 06     42            BNE  OKAY
FF14: 8E EA 03  43            STX  ERFL
FF17: 4C 4F FF  44            JMP  RETURNF     ;Fehler!
FF1A: 48        45   OKAY     PHA
FF1B: 20 C1 FF  46            JSR  SELECT      ;LC-Bank
FF1E: AD EB 03  47            LDA  RW          ;R/W?
FF21: 6A        48            ROR  A
FF22: B0 34     49            BCS  TOBUFF      ;Read
FF24: 68        50            PLA              ;Adreßbyte
FF25: 8D 02 C0  51            STA  RAMRDM      ;RDMAIN
FF28: 8D 05 C0  52            STA  RMWRTA      ;WRAUX
FF2B: 85 07     53            STA  IND2+1      ;Adreßbyte
FF2D: A9 08     54            LDA  #>BUFF      ;Pufferadresse
FF2F: 85 CF     55            STA  IND1+1
                56
FF31: A0 00     57   ACCESS   LDY  #0          ;256 Bytes
FF33: AD E3 FF  58            LDA  STRACK      ;
FF36: C9 0F     59            CMP  #15         ;Spur 15?
FF38: D0 09     60            BNE  NEXT        ;nein
FF3A: 8D 02 C0  61            STA  RAMRDM      ;RDMAIN
FF3D: 8D 04 C0  62            STA  RMWRTM      ;WRMAIN
FF40: 4C 97 0C  63            JMP  DRIVER2     ;Kopieren
FF43: B1 CE     64   NEXT     LDA  (IND1),Y
FF45: 91 06     65            STA  (IND2),Y
FF47: 88        66            DEY
FF48: D0 F9     67            BNE  NEXT        ;fertig?
                68
                69   * Normaler Return
                70
FF4A: A9 00     71            LDA  #00
FF4C: 8D EA 03  72            STA  ERFL        ;CP/M-Fehler
FF4F: 8D 02 C0  73   RETURNF  STA  RAMRDM      ;RDMAIN
FF52: 8D 04 C0  74            STA  RMWRTM      ;WRMAIN
FF55: 4C 90 0C  75            JMP  RTSDRIV     ;zurück
                76
FF58: 68        77   TOBUFF   PLA              ;Adreßbyte
FF59: 8D 03 C0  78            STA  RAMRDA      ;RDAUX
FF5C: 8D 04 C0  79            STA  RMWRTM      ;WRMAIN
FF5F: 85 CF     80            STA  IND1+1      ;Adreßbyte
FF61: A9 08     81            LDA  #>BUFF      ;Pufferadresse
FF63: 85 07     82            STA  IND2+1      ;für Kopieren
FF65: 4C 31 FF  83            JMP  ACCESS      ;Kopieren
                84   *
                85   * CALCADR berechnet die Speicher-Page,
                86   * die dem Track/Sector entspricht.
                87   *
FF68: A2 01     88   CALCADR  LDX  #1
FF6A: A9 00     89            LDA  #0
FF6C: 8D BF FF  90            STA  ADR         ;LSB immer 0
                91   *
FF6F: AD E0 03  92            LDA  TRACK
FF72: C9 0B     93            CMP  #11
FF74: 10 0E     94            BPL  LAB1
FF76: 0A        95            ASL
FF77: 0A        96            ASL
FF78: 0A        97            ASL
FF79: 0A        98            ASL
FF7A: 09 08     99            ORA  #08
FF7C: 18        100  EXIT1    CLC
FF7D: 6D E1 03  101           ADC  SECTOR
FF80: 8D C0 FF  102  EXIT2    STA  ADR+1
FF83: 60        103           RTS
                104  *
FF84: C9 0F     105  LAB1     CMP  #15
FF86: F0 10     106           BEQ  TR15
FF88: C9 0E     107           CMP  #14
FF8A: F0 19     108           BEQ  TR14
FF8C: C9 0D     109           CMP  #13
FF8E: F0 10     110           BEQ  TR13
FF90: C9 0C     111           CMP  #12
FF92: F0 07     112           BEQ  TR12
FF94: C9 0B     113           CMP  #11
FF96: D0 1E     114           BNE  ERR1        ;unzulässig
FF98: 8E BF FF  115  TR15     STX  ADR         ;Bank1 merken
FF9B: A9 D0     116  TR12     LDA  #$D0
FF9D: 4C 7C FF  117           JMP  EXIT1
FFA0: A9 E0     118  TR13     LDA  #$E0
FFA2: 4C 7C FF  119           JMP  EXIT1
FFA5: AD E1 03  120  TR14     LDA  SECTOR
FFA8: C9 0F     121           CMP  #$0F
FFAA: D0 05     122           BNE  LAB2
FFAC: A9 03     123           LDA  #$03
FFAE: 4C 80 FF  124           JMP  EXIT2
FFB1: A9 F0     125  LAB2     LDA  #$F0
FFB3: 4C 7C FF  126           JMP  EXIT1
                127
FFB6: A9 00     128  ERR1     LDA  #00         ;Fehler
FFB8: 8D BF FF  129           STA  ADR
FFBB: 8D C0 FF  130           STA  ADR+1       ;Adresse=0
FFBE: 60        131           RTS              ;zurück
                132
FFBF: 00        133  ADR      HEX  00
FFC0: 00        134           HEX  00
                135
                136  * Select setzt Softswitch für
                137  * Banking der Language-Card
                138
FFC1: AD C0 FF  139  SELECT   LDA  ADR+1
FFC4: 29 F0     140           AND  #%11110000
FFC6: C9 D0     141           CMP  #$D0
FFC8: F0 01     142           BEQ  LCCARD
FFCA: 60        143           RTS
                144
FFCB: AD BF FF  145  LCCARD   LDA  ADR
FFCE: F0 0C     146           BEQ  LBK2
FFD0: A9 00     147           LDA  #00
FFD2: 8D BF FF  148           STA  ADR         ;Bank1-Flag
FFD5: AD 8B C0  149           LDA  BANK1       ;R/W Bank1
FFD8: AD 8B C0  150           LDA  BANK1
FFDB: 60        151           RTS
                152  *
FFDC: AD 83 C0  153  LBK2     LDA  BANK2       ;R/W Bank2
FFDF: AD 83 C0  154           LDA  BANK2
FFE2: 60        155           RTS
                156  *
                157  RTSDRIV  EQU  $0C90
                158  DRIVER2  EQU  $0C97
FFE3: 00        159  STRACK   HEX  00

228 Bytes
```

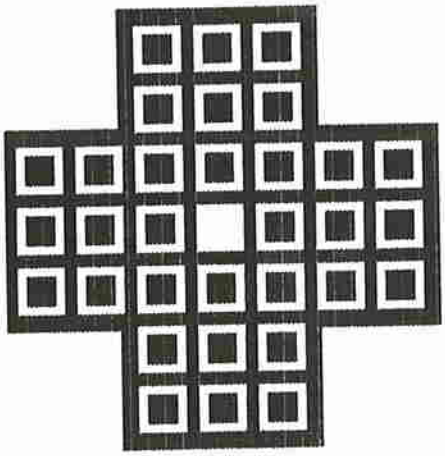
MOVEDRIVER

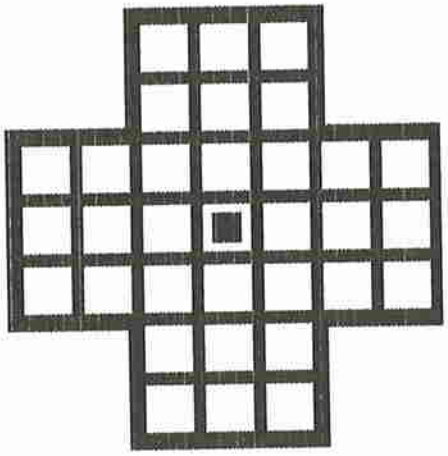
```
                 ***************************
                 *                         *
                 * MOVEDRIVER   AUX $FFØØ  *
                 *                         *
                 ***************************

                          ORG  $95ØØ

                 START    EQU  $ØØ
                 ZIEL     EQU  $Ø2
                 ALTZP    EQU  $CØØ8

95ØØ: 8D Ø9 CØ            STA  ALTZP+1     ;AUXZP
95Ø3: A9 ØØ               LDA  #<SADR
95Ø5: 85 ØØ               STA  START
95Ø7: A9 9Ø               LDA  #>SADR
95Ø9: 85 Ø1               STA  START+1
95ØB: A9 ØØ               LDA  #<DADR
95ØD: 85 Ø2               STA  ZIEL
95ØF: A9 FF               LDA  #>DADR
9511: 85 Ø3               STA  ZIEL+1

9513: AD 8B CØ            LDA  $CØ8B       ;Bank1 R/W
9516: AD 8B CØ            LDA  $CØ8B
9519: AE 33 95            LDX  BLKS
951C: AØ ØØ               LDY  #Ø          ;256 Bytes
951E: B1 ØØ      NEXT     LDA  (START),Y
952Ø: 91 Ø2               STA  (ZIEL),Y
9522: 88                  DEY
9523: DØ F9               BNE  NEXT
9525: E6 Ø1               INC  START+1
9527: E6 Ø3               INC  ZIEL+1
9529: CA                  DEX
952A: DØ F2               BNE  NEXT
952C: 8D Ø8 CØ            STA  ALTZP       ;MAINZP
952F: AD 81 CØ            LDA  $CØ81       ;Read ROM
9532: 6Ø                  RTS

                 SADR     EQU  $9ØØØ
                 DADR     EQU  $FFØØ
9533: Ø1         BLKS     HEX  Ø1

52 Bytes
```

RAMDISK.FORMATTER

```
                 ****************************
                 *                          *
                 * RAMDISK.FORMATTER        *
                 *                          *
                 * Ø3ØØ-Ø3FF                *
                 * Ø8ØØ-BFFF                *
                 * DØØØ-DFFF *2             *
                 * DØØØ-DFFF MAIN           *
                 * EØØØ-FEFF                *
                 *                          *
                 ****************************
                 *
                 BANK2    EQU  $CØ83
                 BANK1    EQU  $CØ8B
                 ALTZPM   EQU  $CØØ8
                 ALTZPA   EQU  $CØØ9
                 RMWRTA   EQU  $CØØ5
                 RMWRTM   EQU  $CØØ4
                 DADR     EQU  $CE
                 *
                          ORG  $9ØØØ
                          OBJ  $9ØØØ
                 *
                 *AUXMAIN -- Ø8ØØ-BFFF
                 *
9ØØØ: 8D Ø5 CØ            STA  RMWRTA
9ØØ3: A9 Ø8               LDA  #Ø8
9ØØ5: 85 CF               STA  DADR+1
9ØØ7: A9 ØØ               LDA  #ØØ
9ØØ9: 85 CE               STA  DADR
9ØØB: A2 B8               LDX  #184
9ØØD: 2Ø 66 9Ø            JSR  INIT
                 *
                 *AUXMAIN -- Ø3ØØ-Ø3FF
                 *
9Ø1Ø: A9 Ø3               LDA  #Ø3
9Ø12: 85 CF               STA  DADR+1
9Ø14: A9 ØØ               LDA  #ØØ
9Ø16: 85 CE               STA  DADR
9Ø18: A2 Ø1               LDX  #1
9Ø1A: 2Ø 66 9Ø            JSR  INIT
                 *
                 *LC-BANK1 -- DØØØ-DFFF/MAIN
                 *
9Ø1D: AD 8B CØ            LDA  BANK1
9Ø2Ø: AD 8B CØ            LDA  BANK1
9Ø23: A9 DØ               LDA  #$DØ
9Ø25: 85 CF               STA  DADR+1
9Ø27: A9 ØØ               LDA  #ØØ
9Ø29: 85 CE               STA  DADR
9Ø2B: A2 1Ø               LDX  #16
9Ø2D: 2Ø 66 9Ø            JSR  INIT
                 *
                 *LC-BANK1,2 -- DØØØ-DFFF/AUX
                 *
9Ø3Ø: AD 8B CØ            LDA  BANK1
9Ø33: AD 8B CØ            LDA  BANK1
9Ø36: 2Ø 3F 9Ø            JSR  INITB
9Ø39: AD 83 CØ            LDA  BANK2
9Ø3C: AD 83 CØ            LDA  BANK2
9Ø3F: 8D Ø9 CØ   INITB    STA  ALTZPA
9Ø42: A9 DØ               LDA  #$DØ
9Ø44: 85 CF               STA  DADR+1
9Ø46: A9 ØØ               LDA  #$ØØ
9Ø48: 85 CE               STA  DADR
9Ø4A: A2 1Ø               LDX  #16
9Ø4C: 2Ø 66 9Ø            JSR  INIT
                 *
                 *COMMON LC -- EØØØ-FEFF
                 *
9Ø4F: A9 EØ               LDA  #$EØ
9Ø51: 85 CF               STA  DADR+1
9Ø53: A9 ØØ               LDA  #$ØØ
9Ø55: 85 CE               STA  DADR
9Ø57: A2 1F               LDX  #31
9Ø59: 2Ø 66 9Ø            JSR  INIT
9Ø5C: 8D Ø4 CØ            STA  RMWRTM
9Ø5F: AD 81 CØ            LDA  $CØ81       ;Read ROM
9Ø62: 8D Ø8 CØ            STA  ALTZPM
9Ø65: 6Ø                  RTS

                 * Init mit $E5, Anzahl der 256-Byte-Blöcke
                 * im X-Register

9Ø66: AØ ØØ      INIT     LDY  #Ø
9Ø68: A9 E5               LDA  #$E5
9Ø6A: 91 CE      NEXT     STA  (DADR),Y
9Ø6C: 88                  DEY
9Ø6D: DØ FB               BNE  NEXT
9Ø6F: E6 CF               INC  DADR+1
9Ø71: CA                  DEX
9Ø72: DØ F6               BNE  NEXT
9Ø74: 6Ø                  RTS

117 Bytes
```

Der nächste Peeker
Heft 7/1985
erscheint am
24. 6. 1985





von Stefan Maas

SOLITAIRE

und wie man es löst

Solitaire ist ein Spiel, bei dem es darum geht, durch Überspringen von Steinen mit anderen Steinen das Spielfeld bis auf einen zu leeren. Dazu ist anfänglich die mittlere Position auf dem Spielfeld unbesetzt. Ein Stein darf nur springen, wenn direkt links, rechts, über oder unter ihm ein Stein steht und die übernächste Position in dieser Richtung unbesetzt ist. Der Stein springt dann auf den freien Platz, wobei der übersprungene Stein entfernt wird. Nach einigen Spielzügen können die verbleibenden Steine so weit entfernt sein, daß sie sich nicht mehr überspringen können. Genau hier liegt die Schwierigkeit von Solitaire.

Das vorliegende Programm **SOLITAIRE** versucht dafür eine Lösung zu finden. Systematisch führt es mögliche Sprünge in der Reihenfolge links, oben, rechts oder unten aus. Gerät es dabei in eine Sackgasse, so macht es den letzten Zug rückgängig und versucht eine andere Möglichkeit. Hält man diese Systematik ein, kann man sicher sein, einen erfolgreichen Spielverlauf zu berechnen. Diese Vorgehensweise des systematischen Programmierens wurde in CHIP Special II als „Backtracking" beschrieben, wobei Programme für 6800 und den TRS 80 vorgestellt wurden.
Auf der Suche nach der Lösung des Solitaire-Problems ist der einzelne Suchschritt relativ einfach (Die Zeilenzahlen beziehen sich auf das Applesoft-Programm SOLITAIRE):

– Nach der Festlegung der Grundstellung (Zeile 160–165) beginnt jeder Versuch beim ersten Stein unten links auf dem Spielbrett. Es werden die Sprungmöglichkeiten in der Reihenfolge links, oben, rechts oder unten geprüft und ausgewertet (Zeile 205–335), gegebenenfalls auch ausgeführt (Zeile 355–420).
– Gibt es keine weiteren Zugmöglichkeiten mehr und ist nicht genau ein Stein im Zentrum des Spielfeldes stehengeblieben, so ist das Spiel in eine Sackgasse geraten. Der letzte Zug wird nun rückgängig gemacht (Zeile 440–460) und nach anderen Zugmöglichkeiten gesucht.
– Die Lösung ist gefunden worden: Ein Stein ist genau im Zentrum des Spielfeldes stehengeblieben. (Zeile 405–410: Es sind X = 31 Züge gemacht worden und der letzte Spielstein besetzt das zentrale Spielfeld A(60) = 1.)
Die Ausgabe der Lösung erfolgt nun entweder über den Drucker (angepaßt an den Drucker Epson MX-82F/T Typ III; ansonsten sind in den Zeilen 610–800 kleine Modifikationen notwendig) oder aber über den Bildschirm (Zeile 560–590).

Für die Grafik sorgen die Routinen in den Zeilen 480–555.

Da das Programm in Applesoft für die Lösung 14 Std. und etwa 35 Min. benötigt, besteht die Möglichkeit der Spielstandabspeicherung (Zeile 400, 820–825), die mit Ctrl-Q aufgerufen wird. Die Zahlenwerte der einzelnen Brettpositionen können der **Tabelle 1** entnommen werden.
Zusätzlich ist auch noch die Assemblerroutine **SOLITAIRE.B** eingebaut (Aufruf in Zeile 945–960), welche die Lösung in 41 Sekunden findet (eine Spielstandabspeicherung ist hierbei nicht möglich) und vom Menü des Applesoft-Programms aus gestartet werden kann. Die Arbeitsweise des Binär-Programms ist mit der des Basic-Programms identisch.

Zusatzinformation: Vor der Eingabe und dem späteren Aufruf des Applesoft-Programms sollte man die folgenden Parameter auf der Zero-Page ändern, da es wegen der Länge im HGR-Speicher liegt: $0067 = $00, $0068 = $40, $3FFF = $00 bzw. von Applesoft aus: POKE 103,0: POKE 104,64: POKE 16383,0. Das Programm **SOLITAIRE.START** führt diese POKEs aus und ruft dann das eigentliche Programm SOLITAIRE auf.
Der Binärfile ist von $8000–$8219 abzuspeichern (BSAVE SOLITAIRE.B, A$8000, L$021A).
Ansonsten erklärt sich das Programm durch sein Menü von selbst.

SOLITAIRE.START

```
10 POKE 103,0: POKE 104,64: POKE 16383,0
20 PRINT CHR$ (4);"RUN SOLITAIRE"
```

SOLITAIRE

```
100 REM ***************************************************
105 REM *                                                 *
110 REM *  SOLITAIRE - von Stefan Maas, MS den 25.10.84   *
115 REM *                                                 *
120 REM ***************************************************
125 REM
130 REM
140 TEXT : HOME : DIM A(120),B(31),C(31):D$(1) = "LINKS ":
    D$(2) = "HOCH  ":D$(3) = "RECHTS":D$(4) = "RUNTER":
    GOTO 845
145 REM ----------------------------
150 REM Festlegung der Grundstellung
155 REM ----------------------------
160 FOR I = 1 TO 32: READ A:A(A) = 1: NEXT I: RESTORE
165 DATA 26,27,28,37,38,39,46,47,48,49,50,51,52,57,58,59,
    61,62,63,68,69,70,71,72,73,74,81,82,83,92,93,94
170 IF P2 = 4 THEN GOTO 565
175 IF P3 = 2 THEN GOTO 825
180 IF P2 = 1 THEN GOTO 610
185 REM ----------------------------------------------------
190 REM Ausführung der Züge
195 REM 1,2,3,4 = l.,o.,r.,u. Bewegung in B/C(1-31) abgesp.
200 REM ----------------------------------------------------
205 I = 25
210 I = I + 1: IF I > 94 THEN GOTO 440: REM kein Zug mehr
    möglich
215 IF A(I) = 0 THEN GOTO 210
220 REM --------------------
225 REM Bewegung nach links
230 REM --------------------
235 IF A(I - 1) = 0 OR A(I - 2) < > 0 OR I = 46 OR I = 47
    OR I = 57 OR I = 58 OR I = 68 OR I = 69 THEN GOTO 260
240 A = I - 2:B = 1: GOSUB 335
245 REM --------------------
250 REM Bewegung nach oben
255 REM --------------------
260 IF A(I + 11) = 0 OR A(I + 22) < > 0 OR I > 80 AND I <
    84 OR I > 91 AND I < 95 THEN GOTO 285
265 A = I + 22:B = 2: GOSUB 335
270 REM --------------------
275 REM Bewegung nach rechts
280 REM --------------------
285 IF A(I + 1) = 0 OR A(I + 2) < > 0 OR I = 51 OR I = 52
    OR I = 62 OR I = 63 OR I = 73 OR I = 74 THEN GOTO 310
290 A = I + 2:B = 3: GOSUB 335
295 REM --------------------
300 REM Bewegung nach unten
305 REM --------------------
310 IF A(I - 11) = 0 OR A(I - 22) < > 0 OR I > 25 AND I <
    29 OR I > 36 AND I < 40 THEN GOTO 210
315 A = I - 22:B = 4: GOSUB 335: GOTO 210
320 REM ------------------------
325 REM Auswertung der Versuche
330 REM ------------------------
335 IF A = 24 OR A = 25 OR A = 29 OR A = 30 OR A = 35 OR A
    = 36 OR A = 40 OR A = 41 OR A = 79 OR A = 80 OR A = 84
    OR A = 85 OR A = 90 OR A = 91 OR A = 95 OR A = 96 THEN
    RETURN
340 REM ------------------------
345 REM Ausführung der Versuche
350 REM ------------------------
355 POP :X = X + 1: REM Pointer
360 Y = Y + 1: VTAB 21: HTAB 10: PRINT Y;
365 VTAB 23: HTAB 11: PRINT I" / "D$(B);
370 A(I) = 0: IF B = 1 THEN HCOLOR= 0:I1 = I: GOSUB 515:
    HCOLOR= 7:I1 = I - 2: GOSUB 515: HCOLOR= 0:I1 = I - 1:
    GOSUB 515:A(I - 1) = 0:A(I - 2) = 1:C(X) = I - 2: GOTO
    390
375 IF B = 2 THEN HCOLOR= 0:I1 = I: GOSUB 515: HCOLOR=
    7:I1 = I + 22: GOSUB 515: HCOLOR= 0:I1 = I + 11: GOSUB
    515:A(I + 11) = 0:A(I + 22) = 1:C(X) = I + 22: GOTO 390
380 IF B = 3 THEN HCOLOR= 0:I1 = I: GOSUB 515: HCOLOR=
    7:I1 = I + 2: GOSUB 515: HCOLOR= 0:I1 = I + 1: GOSUB
    515:A(I + 1) = 0:A(I + 2) = 1:C(X) = I + 2: GOTO 390
385 HCOLOR= 0:I1 = I: GOSUB 515: HCOLOR= 7:I1 = I - 22:
    GOSUB 515: HCOLOR= 0:I1 = I - 11: GOSUB 515:A(I - 11) =
    0:A(I - 22) = 1:C(X) = I - 22
390 IF P3 = 2 THEN GOTO 555
395 IF P2 = 4 THEN GOTO 585
400 B(X) = B:Q = PEEK (49152): IF Q = 145 THEN POKE
    49168,0: GOTO 820
405 IF X < > 31 THEN GOTO 205
410 IF A(60) < > 1 THEN GOTO 205
415 IF P2 = 2 THEN GOTO 560
420 IF P2 = 1 THEN GOTO 710
425 REM --------------------
430 REM Rüchnahme von Zügen
435 REM --------------------
440 A(C(X)) = 0
445 IF B(X) = 1 THEN I = C(X) + 2: HCOLOR= 0:I1 = C(X):
    GOSUB 515: HCOLOR= 7:I1 = I - 1: GOSUB 515:I1 = I:
    GOSUB 515:A(C(X) + 1) = 1:A(C(X) + 2) = 1:X = X - 1:
    GOTO 260
450 IF B(X) = 2 THEN I = C(X) - 22: HCOLOR= 0:I1 = C(X):
    GOSUB 515: HCOLOR= 7:I1 = I + 11: GOSUB 515:I1 = I:
    GOSUB 515:A(C(X) - 11) = 1:A(C(X) - 22) = 1:X = X - 1:
    GOTO 285
455 IF B(X) = 3 THEN I = C(X) - 2: HCOLOR= 0:I1 = C(X):
    GOSUB 515: HCOLOR= 7:I1 = I + 1: GOSUB 515:I1 = I:
    GOSUB 515:A(C(X) - 1) = 1:A(C(X) - 2) = 1:X = X - 1:
    GOTO 310
460 I = C(X) + 22: HCOLOR= 0:I1 = C(X): GOSUB 515: HCOLOR=
    7:I1 = I - 11: GOSUB 515:I1 = I: GOSUB 515:A(C(X) + 11)
    = 1:A(C(X) + 22) = 1:X = X - 1: GOTO 210
465 REM ----------------
470 REM Grafik mit Text
475 REM ----------------
480 HOME : HGR : HCOLOR= 7
485 HPLOT 116,42 TO 162,42 TO 162,70 TO 190,70 TO 190,116
    TO 162,116 TO 162,144 TO 116,144 TO 116,116 TO 88,116
    TO 88,70 TO 116,70 TO 116,42
490 X = 117:Y = 43: GOSUB 495:X = 89:Y = 71: GOSUB 495:X =
    145:Y = 71: GOSUB 495:X = 117:Y = 99: GOSUB 495: GOTO
    505
495 FOR I = 1 TO 4: FOR J = 0 TO 2: HPLOT X,Y + J TO X +
    45,Y + J: NEXT J:Y = Y + 14: NEXT I:Y = Y - 56
500 FOR I = 1 TO 4: FOR J = 0 TO 2: HPLOT X + J,Y TO X +
    J,Y + 45: NEXT J:X = X + 14: NEXT I: RETURN
505 FOR I1 = 26 TO 28: GOSUB 515: NEXT I1: FOR I1 = 37 TO
    39: GOSUB 515: NEXT I1: FOR I1 = 46 TO 52: GOSUB 515:
    NEXT I1: FOR I1 = 57 TO 59: GOSUB 515: NEXT I1: FOR I1
    = 61 TO 63: GOSUB 515: NEXT I1: FOR I1 = 68 TO 74:
    GOSUB 515: NEXT I1
510 FOR I1 = 81 TO 83: GOSUB 515: NEXT I1: FOR I1 = 92 TO
    94: GOSUB 515: NEXT I1: GOTO 525
515 I2 = I1 / 11:X1 = INT ((I2 - INT (I2)) * 10) * 14 +
    80:Y1 = 160 - INT (I2) * 14
520 FOR J = 0 TO 6: HPLOT X1,Y1 + J TO X1 + 6,Y1 + J: NEXT
    J: RETURN
525 VTAB 21: PRINT "Versuch:": VTAB 23: PRINT "Position:";
530 X = 0:Y = 0: GOTO 160
535 T = X:Y = Y - X: FOR X = 1 TO T:B = B(X): IF B = 1 THEN
    I = C(X) + 2: GOTO 360
540 IF B = 2 THEN I = C(X) - 22: GOTO 360
545 IF B = 3 THEN I = C(X) - 2: GOTO 360
550 I = C(X) + 22: GOTO 360
555 NEXT X:P3 = 0:X = X - 1: GOTO 205
560 T = X:T1 = Y:P2 = 4: GOSUB 480
565 Y = T1 - 31: FOR X = 1 TO T: GOSUB 590:B = B(X): IF B =
    1 THEN I = C(X) + 2: GOTO 360
570 IF B = 2 THEN I = C(X) - 22: GOTO 360
575 IF B = 3 THEN I = C(X) - 2: GOTO 360
580 I = C(X) + 22: GOTO 360
585 NEXT X: GOSUB 590:X = X - 1: TEXT : HOME : END
590 VTAB 23: HTAB 28: PRINT "Taste:";: GET T$: RETURN
595 REM --------------------
600 REM Druckeransteuerung
605 REM --------------------
610 VTAB 10: POKE 35,5: PR# 1
615 PRINT CHR$ (27);"D"; CHR$ (34); CHR$ (38); CHR$ (0)
620 PRINT CHR$ (14); CHR$ (9);"--------------"
625 PRINT CHR$ (14); CHR$ (9); CHR$ (9);"SOLITAIRE"
630 PRINT CHR$ (27);"S"; CHR$ (0); CHR$ (15);"
    Geschrieben von St. Maas, Münster 25.10.84"
635 PRINT CHR$ (27);"H"; CHR$ (27);"T"; CHR$ (18);
640 GOSUB 800: IF U = 1 THEN U = 0: GOTO 715
645 GOSUB 790: PRINT "Brettkodierung:";: GOSUB 795
650 PRINT CHR$ (27);"D"; CHR$ (48); CHR$ (0);
655 GOSUB 790: PRINT CHR$ (27);"D"; CHR$ (33); CHR$ (0);
    CHR$ (9);"Zug-Beispiel:";: GOSUB 795
660 PRINT CHR$ (27);"D"; CHR$ (48); CHR$ (0)
665 PRINT : PRINT "       92 93 94"; CHR$ (0); CHR$ (9);
    "38 / HOCH bedeutet:"
670 PRINT "       81 82 83"
675 PRINT "68 69 70 71 72 73 74"; CHR$ (0); CHR$ (9);"Der
    Stein auf Feld 38 wird auf Feld 60 (HOCH)"
680 PRINT "57 58 59 60 61 62 63"; CHR$ (0); CHR$
    (9);"gesetzt und der übersprungende Stein (Feld 49)"
```

```
685 PRINT "46 47 48 49 5Ø 51 52"; CHR$ (Ø); CHR$
    (9);"entfernt."
69Ø PRINT "      37 38 39"
695 PRINT "      26 27 28": PRINT
7ØØ GOSUB 8ØØ: PR# Ø: POKE 35,24: IF P1 = 1 THEN GOTO 71Ø
7Ø5 GOTO 2Ø5
71Ø VTAB 1Ø: POKE 35,5:L = Ø: PR# 1: IF U = 1 THEN GOTO
    615
715 PRINT CHR$ (13);"Lösung:"
72Ø PRINT CHR$ (27);"D"; CHR$ (1); CHR$ (21); CHR$ (41);
    CHR$ (61); CHR$ (81); CHR$ (Ø)
725 GOSUB 78Ø
73Ø FOR X = 1 TO 3Ø: PRINT CHR$ (9);: IF X < 1Ø THEN
    PRINT " ";: GOTO 76Ø
735 GOTO 76Ø
74Ø IF B(X) = 1 THEN D = C(X) + 2: RETURN
745 IF B(X) = 2 THEN D = C(X) - 22: RETURN
75Ø IF B(X) = 3 THEN D = C(X) - 2: RETURN
755 D = C(X) + 22: RETURN
76Ø GOSUB 74Ø: PRINT X;". ";D;" / ";D$(B(X));"  I";:L = L +
    1: IF L < > 4 THEN GOTO 775
765 L = Ø:X = X + 1: PRINT CHR$ (9);: IF X < 1Ø THEN
    PRINT " ";
77Ø GOSUB 74Ø: PRINT X;". ";D;" / ";D$(B(X)); CHR$ (Ø)
775 NEXT X: GOSUB 74Ø: PRINT CHR$ (9);X;". ";D;" /
    ";D$(B(X)): GOSUB 8ØØ: GOSUB 785: PR# Ø: TEXT : HOME :
    END
78Ø PRINT CHR$ (27);"R"; CHR$ (Ø);: RETURN : REM
    USA Character-Mode ein
785 PRINT CHR$ (27);"R"; CHR$ (2);: RETURN : REM
    BRD Character-Mode ein
79Ø PRINT CHR$ (27);"-"; CHR$ (1);: RETURN : REM
    Underline-Mode ein
795 PRINT CHR$ (27);"-"; CHR$ (Ø);: RETURN : REM
    Underline-Mode aus
8ØØ PRINT CHR$ (14);: FOR I = 1 TO 48: PRINT "-";: NEXT I:
    PRINT : PRINT : RETURN : REM Zeilenabgrenzung
8Ø5 REM ----------------------
81Ø REM Spielstandabspeicherung
815 REM ----------------------
82Ø PRINT : PRINT CHR$ (4);"OPEN SOLITAIRE.SPIELSTAND":
    PRINT CHR$ (4);"WRITE SOLITAIRE.SPIELSTAND": PRINT X:
    PRINT Y: FOR J = 1 TO X: PRINT B(J): PRINT C(J): NEXT
    J: PRINT CHR$ (4);"CLOSE SOLITAIRE.SPIELSTAND": TEXT :
    HOME : END
825 PRINT : PRINT CHR$ (4);"OPEN SOLITAIRE.SPIELSTAND":
    PRINT CHR$ (4);"READ SOLITAIRE.SPIELSTAND": INPUT X,Y:
    FOR J = 1 TO X: INPUT B(J),C(J): NEXT J: PRINT CHR$
    (4);"CLOSE SOLITAIRE.SPIELSTAND":U = 1: GOTO 535
83Ø REM ----
835 REM Menü
84Ø REM ----
845 INVERSE : PRINT "----------------------------------------
    SOLITAIRE von Stefan Maas   25.1Ø.84
    ----------------------------------------": NORMAL
85Ø PRINT "Lösungsmöglichkeiten:": PRINT : HTAB 3: PRINT
    "(1) schnelle Lösung (Ausführungsdauer": HTAB 7: PRINT
    "ca. 41 Sekunden, ohne Grafik)": PRINT
855 HTAB 3: PRINT "(2) langsame Lösung (Ausführungsdauer":
    HTAB 7: PRINT "ca. 14,5 Stunden, mit Grafik)": PRINT
86Ø PRINT "Auagabe der gefundenen Lösungen:": PRINT : HTAB
    3: PRINT "(3) auf Drucker";: HTAB 22: PRINT "(4) auf
    Bildschirm": PRINT
865 PRINT "Programmstart:": PRINT : HTAB 3: PRINT "(5)
    Neustart": HTAB 3: PRINT "(6) unterbrochene Partie neu
    starten"
87Ø POKE 1795,49: POKE 1835,51: POKE 1363,53:P1 = 1:P2 =
    1:P3 = 1
875 VTAB 24: HTAB 1: PRINT "Wähle 1-6 (RETURN <=> Start):";
88Ø GET P$: IF ASC (P$) = 13 AND P1 = 2 THEN GOTO 48Ø
885 IF ASC (P$) = 13 THEN GOTO 945
89Ø IF ASC (P$) < 49 OR ASC (P$) > 54 THEN GOTO 875
895 P = VAL (P$): ON P GOTO 9ØØ,9Ø5,91Ø,915,92Ø,925
9ØØ POKE 1795,49: POKE 1195,178:P1 = 1: GOTO 92Ø
9Ø5 POKE 1195,5Ø: POKE 1795,177:P1 = 2: GOTO 88Ø
91Ø POKE 1835,51: POKE 1854,18Ø:P2 = 1: GOTO 88Ø
915 POKE 1854,52: POKE 1835,179:P2 = 2: GOTO 88Ø
92Ø POKE 1363,53: POKE 1491,182:P3 = 1: GOTO 88Ø
925 POKE 1491,54: POKE 1363,181:P3 = 2: GOTO 9Ø5
93Ø REM ------------------
935 REM Binärfilesteuerung
94Ø REM ------------------
945 HOME : HTAB 4: PRINT "SOLITAIRE-Binärfile wird
    geladen": PRINT CHR$ (4);"BLOAD SOLITAIRE.B": VTAB 5:
    HTAB 9: PRINT "Lösungssuche gestartet": PRINT : HTAB 2:
    PRINT "(Lösung bei Klinkelzeichen gefunden)": CALL
    32768
95Ø VTAB 11: HTAB 13: PRINT "Datenübergabe": FOR I = 1 TO
    31:B(I) = PEEK (33631 + I):C(I) = PEEK (33663 + I):
    NEXT I
955 IF P2 = 1 THEN GOTO 61Ø
96Ø Y = 256 * PEEK (251) + PEEK (252):X = 31: GOTO 56Ø
```

SOLITAIRE.B

```
1    ***********************************
2    * SOLITAIRE - St. Maas, 25.1Ø.84 *
3    ***********************************
4
5              ORG  $8ØØØ
6    POINTER1 EQU  $FF
7    POINTER2 EQU  $FE
8    POINTER3 EQU  $FD
9    VERSL    EQU  $FC
1Ø   VERSH    EQU  $FB
11   SPIEL    EQU  $83ØØ
12
13   ************************
14   * Spielfelddefinierung *
15   ************************
8ØØØ: A9 ØØ      16   START    LDA  #ØØ
8ØØ2: 85 FD      17            STA  POINTER3
8ØØ4: 85 FC      18            STA  VERSL
8ØØ6: 85 FB      19            STA  VERSH
8ØØ8: A2 78      2Ø            LDX  #12Ø
8ØØA: CA         21   SPIEL1   DEX
8ØØB: 9D ØØ 83   22            STA  SPIEL,X
8ØØE: DØ FA      23            BNE  SPIEL1
81Ø: A9 Ø1       24            LDA  #Ø1
8Ø12: BC 2Ø 8Ø   25   SPIEL2   LDY  FELD,X
8Ø15: 99 ØØ 83   26            STA  SPIEL,Y
8Ø18: E8         27            INX
8Ø19: EØ 2Ø      28            CPX  #32
8Ø1B: FØ 23      29            BEQ  TEST
8Ø1D: 4C 12 8Ø   3Ø            JMP  SPIEL2
8Ø2Ø: 1A 1B 1C   31   FELD     HEX  1A1B1C2526272E2F
8Ø23: 25 26 27 2E 2F
8Ø28: 3Ø 31 32   32            HEX  3Ø31323334393A3B
8Ø2B: 33 34 39 3A 3B
8Ø3Ø: 3D 3E 3F   33            HEX  3D3E3F4445464748
8Ø33: 44 45 46 47 48
8Ø38: 49 4A 51   34            HEX  494A5152535C5D5E
8Ø3B: 52 53 5C 5D 5E
                 35   ***********************************
                 36   * Ausführung der Züge             *
                 37   * 1,2,3,4 = l.,o.,r.,u. Bewegung *
                 38   ***********************************
8Ø4Ø: AØ 19      39   TEST     LDY  #25
8Ø42: 84 FF      4Ø            STY  POINTER1
8Ø44: A4 FF      41   TEST1    LDY  POINTER1
8Ø46: C8         42            INY
8Ø47: 84 FF      43            STY  POINTER1
8Ø49: CØ 5F      44            CPY  #95
8Ø4B: DØ Ø3      45            BNE  TEST1A
8Ø4D: 4C 9A 81   46            JMP  TESTNEG
8Ø5Ø: B9 ØØ 83   47   TEST1A   LDA  SPIEL,Y
8Ø53: FØ EF      48            BEQ  TEST1
8Ø55: 84 FF      49            STY  POINTER1
                 5Ø   **********************
                 51   * Bewegung nach links *
                 52   **********************
8Ø57: A4 FF      53   LINKS    LDY  POINTER1
8Ø59: 88         54            DEY
8Ø5A: 84 FE      55            STY  POINTER2
8Ø5C: B9 ØØ 83   56            LDA  SPIEL,Y
8Ø5F: FØ 23      57            BEQ  OBEN
8Ø61: 88         58            DEY
8Ø62: B9 ØØ 83   59            LDA  SPIEL,Y
8Ø65: DØ 1D      6Ø            BNE  OBEN
8Ø67: CØ 2C      61            CPY  #44
8Ø69: FØ 19      62            BEQ  OBEN
8Ø6B: CØ 2D      63            CPY  #45
8Ø6D: FØ 15      64            BEQ  OBEN
8Ø6F: CØ 37      65            CPY  #55
8Ø71: FØ 11      66            BEQ  OBEN
8Ø73: CØ 38      67            CPY  #56
8Ø75: FØ ØD      68            BEQ  OBEN
8Ø77: CØ 42      69            CPY  #66
8Ø79: FØ Ø9      7Ø            BEQ  OBEN
8Ø7B: CØ 43      71            CPY  #67
8Ø7D: FØ Ø5      72            BEQ  OBEN
8Ø7F: A2 Ø1      73            LDX  #Ø1
8Ø81: 2Ø 1E 81   74            JSR  TEST2
```

```
                  75   ***********************
                  76   * Bewegung nach oben *
                  77   ***********************
8084: A5 FF       78   OBEN     LDA  POINTER1
8086: 18          79            CLC
8087: 69 0B       80            ADC  #11
8089: A8          81            TAY
808A: 85 FE       82            STA  POINTER2
808C: B9 00 83    83            LDA  SPIEL,Y
808F: F0 28       84            BEQ  RECHTS
8091: A5 FE       85            LDA  POINTER2
8093: 18          86            CLC
8094: 69 0B       87            ADC  #11
8096: A8          88            TAY
8097: B9 00 83    89            LDA  SPIEL,Y
809A: D0 1D       90            BNE  RECHTS
809C: C0 67       91            CPY  #103
809E: F0 19       92            BEQ  RECHTS
80A0: C0 68       93            CPY  #104
80A2: F0 15       94            BEQ  RECHTS
80A4: C0 69       95            CPY  #105
80A6: F0 11       96            BEQ  RECHTS
80A8: C0 72       97            CPY  #114
80AA: F0 0D       98            BEQ  RECHTS
80AC: C0 73       99            CPY  #115
80AE: F0 09       100           BEQ  RECHTS
80B0: C0 74       101           CPY  #116
80B2: F0 05       102           BEQ  RECHTS
80B4: A2 02       103           LDX  #02
80B6: 20 1E 81    104           JSR  TEST2
                  105  ************************
                  106  * Bewegung nach rechts *
                  107  ************************
80B9: A4 FF       108  RECHTS   LDY  POINTER1
80BB: C8          109           INY
80BC: 84 FE       110           STY  POINTER2
80BE: B9 00 83    111           LDA  SPIEL,Y
80C1: F0 23       112           BEQ  UNTEN
80C3: C8          113           INY
80C4: B9 00 83    114           LDA  SPIEL,Y
80C7: D0 1D       115           BNE  UNTEN
80C9: C0 35       116           CPY  #53
80CB: F0 19       117           BEQ  UNTEN
80CD: C0 36       118           CPY  #54
80CF: F0 15       119           BEQ  UNTEN
80D1: C0 40       120           CPY  #64
80D3: F0 11       121           BEQ  UNTEN
80D5: C0 41       122           CPY  #65
80D7: F0 0D       123           BEQ  UNTEN
80D9: C0 4B       124           CPY  #75
80DB: F0 09       125           BEQ  UNTEN
80DD: C0 4C       126           CPY  #76
80DF: F0 05       127           BEQ  UNTEN
80E1: A2 03       128           LDX  #03
80E3: 20 1E 81    129           JSR  TEST2
                  130  ************************
                  131  * Bewegung nach unten *
                  132  ************************
80E6: A5 FF       133  UNTEN    LDA  POINTER1
80E8: 38          134           SEC
80E9: E9 0B       135           SBC  #11
80EB: A8          136           TAY
80EC: 85 FE       137           STA  POINTER2
80EE: B9 00 83    138           LDA  SPIEL,Y
80F1: F0 28       139           BEQ  TEST11
80F3: A5 FE       140           LDA  POINTER2
80F5: 38          141           SEC
80F6: E9 0B       142           SBC  #11
80F8: A8          143           TAY
80F9: B9 00 83    144           LDA  SPIEL,Y
80FC: D0 1D       145           BNE  TEST11
80FE: C0 04       146           CPY  #4
8100: F0 19       147           BEQ  TEST11
8102: C0 05       148           CPY  #5
8104: F0 15       149           BEQ  TEST11
8106: C0 06       150           CPY  #6
8108: F0 11       151           BEQ  TEST11
810A: C0 0F       152           CPY  #15
810C: F0 0D       153           BEQ  TEST11
810E: C0 10       154           CPY  #16
8110: F0 09       155           BEQ  TEST11
8112: C0 11       156           CPY  #17
8114: F0 05       157           BEQ  TEST11
8116: A2 04       158           LDX  #04
8118: 20 1E 81    159           JSR  TEST2
811B: 4C 44 80    160  TEST11   JMP  TEST1
```

```
                  161  ********************************
                  162  * Zusatzauswertung der Versuche *
                  163  ********************************
811E: C0 18       164  TEST2    CPY  #24
8120: F0 3F       165           BEQ  TEST3
8122: C0 19       166           CPY  #25
8124: F0 3B       167           BEQ  TEST3
8126: C0 1D       168           CPY  #29
8128: F0 37       169           BEQ  TEST3
812A: C0 1E       170           CPY  #30
812C: F0 33       171           BEQ  TEST3
812E: C0 23       172           CPY  #35
8130: F0 2F       173           BEQ  TEST3
8132: C0 24       174           CPY  #36
8134: F0 2B       175           BEQ  TEST3
8136: C0 28       176           CPY  #40
8138: F0 27       177           BEQ  TEST3
813A: C0 29       178           CPY  #41
813C: F0 23       179           BEQ  TEST3
813E: C0 4F       180           CPY  #79
8140: F0 1F       181           BEQ  TEST3
8142: C0 50       182           CPY  #80
8144: F0 1B       183           BEQ  TEST3
8146: C0 54       184           CPY  #84
8148: F0 17       185           BEQ  TEST3
814A: C0 55       186           CPY  #85
814C: F0 13       187           BEQ  TEST3
814E: C0 5A       188           CPY  #90
8150: F0 0F       189           BEQ  TEST3
8152: C0 5B       190           CPY  #91
8154: F0 0B       191           BEQ  TEST3
8156: C0 5F       192           CPY  #95
8158: F0 07       193           BEQ  TEST3
815A: C0 60       194           CPY  #96
815C: F0 03       195           BEQ  TEST3
815E: 4C 62 81    196           JMP  TEST4
8161: 60          197  TEST3    RTS
                  198  ************************
                  199  * Ausführung von Zügen *
                  200  ************************
8162: E6 FC       201  TEST4    INC  VERSL
8164: D0 02       202           BNE  VERS1
8166: E6 FB       203           INC  VERSH
8168: 68          204  VERS1    PLA
8169: 68          205           PLA
816A: E6 FD       206           INC  POINTER3
816C: 8A          207           TXA
816D: A6 FD       208           LDX  POINTER3
816F: 9D 5F 83    209           STA  $835F,X
8172: 98          210           TYA
8173: 9D 7F 83    211           STA  $837F,X
8176: A9 01       212           LDA  #01
8178: 99 00 83    213           STA  SPIEL,Y
817B: A9 00       214           LDA  #00
817D: A4 FF       215           LDY  POINTER1
817F: 99 00 83    216           STA  SPIEL,Y
8182: A4 FE       217           LDY  POINTER2
8184: 99 00 83    218           STA  SPIEL,Y
8187: E0 1F       219           CPX  #31
8189: F0 03       220           BEQ  ENDE
818B: 4C 40 80    221           JMP  TEST
818E: AD 3C 83    222  ENDE     LDA  $833C
8191: D0 03       223           BNE  ENDE1
8193: 4C 40 80    224           JMP  TEST
8196: 20 DD FB    225  ENDE1    JSR  $FBDD
8199: 60          226           RTS
                  227  ************************
                  228  * Rücknahme von Zügen *
                  229  ************************
819A: A4 FD       230  TESTNEG  LDY  POINTER3
819C: B9 5F 83    231           LDA  $835F,Y
819F: BE 7F 83    232           LDX  $837F,Y
81A2: C9 01       233           CMP  #01
81A4: F0 28       234           BEQ  TESTNEG1
81A6: C9 02       235           CMP  #02
81A8: F0 3A       236           BEQ  TESTNEG2
81AA: C9 03       237           CMP  #03
81AC: F0 56       238           BEQ  TESTNEG3
81AE: A9 00       239           LDA  #00
81B0: 9D 00 83    240           STA  SPIEL,X
81B3: 8A          241           TXA
81B4: 18          242           CLC
81B5: 69 0B       243           ADC  #11
81B7: AA          244           TAX
81B8: A9 01       245           LDA  #01
81BA: 9D 00 83    246           STA  SPIEL,X
```

```
81BD: 8A          247            TXA
81BE: 18          248            CLC
81BF: 69 ØB       249            ADC  #11
81C1: AA          25Ø            TAX
81C2: A9 Ø1       251            LDA  #Ø1
81C4: 9D ØØ 83    252            STA  SPIEL,X
81C7: C6 FD       253            DEC  POINTER3
81C9: 86 FF       254            STX  POINTER1
81CB: 4C 44 8Ø    255            JMP  TEST1
81CE: A9 ØØ       256  TESTNEG1  LDA  #ØØ
81DØ: 9D ØØ 83    257            STA  SPIEL,X
81D3: A9 Ø1       258            LDA  #Ø1
81D5: E8          259            INX
81D6: 9D ØØ 83    26Ø            STA  SPIEL,X
81D9: E8          261            INX
81DA: 9D ØØ 83    262            STA  SPIEL,X
81DD: C6 FD       263            DEC  POINTER3
81DF: 86 FF       264            STX  POINTER1
81E1: 4C 84 8Ø    265            JMP  OBEN
81E4: A9 ØØ       266  TESTNEG2  LDA  #ØØ
81E6: 9D ØØ 83    267            STA  SPIEL,X
81E9: 8A          268            TXA
81EA: 38          269            SEC
81EB: E9 ØB       27Ø            SBC  #11
81ED: AA          271            TAX
81EE: A9 Ø1       272            LDA  #Ø1
81FØ: 9D ØØ 83    273            STA  SPIEL,X
81F3: 8A          274            TXA
81F4: 38          275            SEC
```

```
81F5: E9 ØB       276            SBC  #11
81F7: AA          277            TAX
81F8: A9 Ø1       278            LDA  #Ø1
81FA: 9D ØØ 83    279            STA  SPIEL,X
81FD: C6 FD       28Ø            DEC  POINTER3
81FF: 86 FF       281            STX  POINTER1
82Ø1: 4C B9 8Ø    282            JMP  RECHTS
82Ø4: A9 ØØ       283  TESTNEG3  LDA  #ØØ
82Ø6: 9D ØØ 83    284            STA  SPIEL,X
82Ø9: A9 Ø1       285            LDA  #Ø1
82ØB: CA          286            DEX
82ØC: 9D ØØ 83    287            STA  SPIEL,X
82ØF: CA          288            DEX
821Ø: 9D ØØ 83    289            STA  SPIEL,X
8213: C6 FD       29Ø            DEC  POINTER3
8215: 86 FF       291            STX  POINTER1
8217: 4C E6 8Ø    292            JMP  UNTEN

538 Bytes
```

Tabelle 1: Brettkodierung

```
      92 93 94
      81 82 83
68 69 7Ø 71 72 73 74
57 58 59 6Ø 61 62 63
46 47 48 49 5Ø 51 52
      37 38 39
      26 27 28
```

Zugbeispiel:

38 / HOCH bedeutet:

Der Stein auf Feld 38 wird auf Feld 6Ø (HOCH) gesetzt und der übersprungene Stein (Feld 49) entfernt.

BÜHLER-LIGHTNING

Katalog kostenlos! Heiße Renner Blitz-Kontakt (07221) 3487

Texas Software Cassetten. Lerne BASIC-1-9 u. 5 versch. Spiele. 2 Cassetten.
BN 93177 **DM 9,90**

ATARI-Sensorkeyboard. Matrix. Tasten frei belegbar. Ausgeprägte Tasten f. sichere Eingabe. 61 Tasten.
BN 93185 **DM 9,80**

ASC II-Keyboards evtl. können einige Tasten fehlen. Wir liefern dafür 2 gleiche Keyboards. Aus 2 mach 1.
BN 93190 2 x ASCII **DM 24,90**
BN 93191 4 Stück nur **DM 45,—**

2 x 3 Amp. Computer Netzfilter
BN 93192 **DM 5,90**

2 x 2 Amp. aber mit Kalgeräte-Steckdose
BN 93194 **DM 7,90**

ASC II-Keyboards mit Gehäuse

ASCII-Standard Tastatur Apple u. NDR Computer fähig. Problemloser Anschluß. Volle Garantie mit Anschluß manual. Pultgehäuse. Industrieware. Kabel mit 26-pol. Stecker. Processor AMI 8207 Stromvers. ±5 V.
BN 93077 **DM 98,—**

Dieselbe Tastatur mit 10er Block u. Cursor. Processor AMI 8021.
BN 93078 **DM 139,—**

Tastatur wie BN 93078 ohne Gehäuse
BN 93079 **DM 89,90**

Der Massenspeicher 1 M-Byte

Mit Anschlußanleitung für Apple-Computer. 3"-Mini-Floppy-Laufwerk. 2 x 80 = 160 Track. 1 Mega-B Doppelkopf. Standard Shugart BUS. Qualitätslaufwerke eines bekannten Japanischen Herstellers. Guß-Chassis. 1 Jahr Garantie. Kundendienst über unsere Organisation. Minimaße nur 150 x 100 x H 40 mm.
BN 65163 1 M-BYTE-KAPAZ. **DM 598,—**

Graphic Computer Mouse

Graphic-Designer Apple compatible. Erstellt Zeichnungen und Graphicen einfach und schnell. Direkt auf den Bildschirm. Gute Auflösung. Viele Funktionen bereits programmiert wie Frame/Line/Circle/Ellype/Fill/Prints u. viel, viel mehr. Komfortable Menu-Steuerung. Anschluß über Game IO Port. Lieferung mit Kabel, Software 5 ¼" Discette und Anleitung.
BN 65156 **DM 189,—**

Unser Kundendienst in Baden-Baden und Karlsruhe kann alles reparieren. Wir verfügen über alle Ersatzteile.

Paßt an jeden Apple und ähnliche

Lautloses 5¼" Floppy-Laufwerk völlig narrensicher. Hochwertigste jap. Mechanik. Elektronik garantiert ohne Ausfälle. Wir geben gerne 1 Jahr Garantie auf diese Qualität. Slim Line. Jap. Spitzentechnologie.
BN 65147 Geräuschloses 5¼" **DM 448,—**
BN 65148 Pass. Slim Line Geh. **DM 38,—**
BN 65011 Disc Drive Card **DM 119,—**

Blitzpreis! Japan-Qualität

5¼"/Apple compatible Anschlußkabel
Apple fähiges Floppy. Slim Line. Metall-Gehäuse. **1 Jahr Vollgarantie.** Sofort anschließbar. Einfach nur am Apple einstecken.
BN 65151 **DM 399,90**

Original Data Magnetics US-Qualität 5¼"
100% Errorfree. Jede defekte Discette tauschen wir gegen 2 neue. 5 Jahre Garantie auf dieses Material.
DM 1 S SS-SD BN 65152 10er Pack **DM 38,—**
DM 1 D SS-DD BN 65153 10er Pack **DM 38,—**
DM 2 D DS-DD BN 65154 10er Pack **DM 42,—**

Hochwertige Interface-Leerplatinen für Apple u. ä.

Epoxy oder glasfaserverstärktes Material. Bereits gebohrt.

65164	64 K Motherboard Dual-Processor	89,—	65171	Eprom Writer	19,50
			65172	PAL/VIDEO/Modul.	19,50
65165	Z-80	19,50	65173	128 K	29,50
65166	80 Zeichen/Soft	19,50	65174	Mikro Buffer	19,90
65167	Serial/RS 232	15,90	65175	Buffer Printer	25,80
65168	Parallel/8 Bit	15,90	65176	Super Serial	19,90
65169	16 K RAM	19,50	65177	A/D	19,50
65170	6522	15,90	65178	Micro Modem	29,50

Soft und Zubehör

5 Stuck Comp. Cassetten. 10 Min. Tonbandspulen.
BN 65045 **DM 9,95**

5 Stuck Comp. Cassetten. 15 Min. ohne Tonbandspulen.
BN 65046 **DM 6,90**

5 Stuck Discettenbox 5¼" fur je 5 Discetten farbig.
BN 65112 **DM 12,90**

5¼" Discetten Bibliothek fur 10 Discetten Kunststoff
BN 65057 **DM 8,90**

5¼" Discetten Bibliothek fur 15 Discetten Kunststoff
BN 65130 **DM 19,80**

5¼" Discetten Bibliothek fur 50 Discetten Kunststoff
BN 65110 **DM 28,50**

5¼" Discetten Bibliothek fur 100 Discetten Kunststoff abschließbar
BN 65049 **DM 49,80**

Disc Doubler (Notcher) macht aus einer SS eine DS 5¼"
BN 65143 **DM 14,90**

*Microswitch. Standard ASCII Parallel. Halltastatur/Cursor/Viele Funktionstasten/+5 V/–12 V.
BN 92272 Apple u. a. **DM 178,—**

Der SUPERKNÜPPEL FÜR APPLE und kompatible. VG-071.
Joystick der edelsten Sorte.
Leichtgängiger Kreuzknuppel mit automatischer Nullpunkt-Ruckstellung. Spezielle Trimmregler zur genauen Einpegelung fur jeden Kanal. 2 Abschußknöpfe. Rutschfeste Gummifüße. Flexibles Anschlußkabel 1,60 m lang. Komplett mit Stecker, sofort betriebsbereit. Metallgehäuse.
BN 65056 VG-071 **DM 49,90**

Original TEAC 55 F. Double Density. 2 x 80 Track.
BN 65179 **DM 598,—**

300 Band Modem CCITTV. 21
Postgenehmigt mit FTZ. Anschluß an alle Computer mit V 24 Schnittstelle. Schnittstellenstecker. Vollduplex. Answer- und Originalmodus. Autom. Kanalwahl. Made in Germany.
BN 65180 **DM 298,—**

Neue Discetten-Boxen

Karussell-Box für 64 Discetten 5¼". Bequemer Zugriff. Stapelbar oder mit Wandhalterung. Ø 320 x H 200 mm.
BN 65155 **DM 59,80**

Jetzt 3,5" Discetten-Box für **50 Stück 3,5" Discetten** Klappbox mit Sortiereinrichtung.
BN 65146 **DM 29,50**

Teilzahlung ab DM 300,– (ohne Zinsen) Schnell-Versand Verpackung frei

Bühler Elektronik, Postfach 32, 7570 Baden-Baden, Tel. (07221) 3487

Leserbriefe

Garbage Collection bei TASC

Zuerst möchte ich mich für die Veröffentlichung dieser wirklich interessanten Routine bedanken, durch welche es möglich ist, die meiner Meinung nach größte Schwachstelle des Applesoft-Interpreters auf einfache und sichere Art zu umgehen.
Des weiteren möchte ich nun anfragen, ob die Möglichkeit besteht, diese Routine auch auf mit TASC 2.0 compilierte Programme anzuwenden, und welche Modifikationen hierzu notwendig sind.
Kurt Lutsch, Esslingen-Zell
Diese Frage wurde uns sehr häufig gestellt. Das FRE-Programm aus Peeker 1/2, 1985, läuft jedoch nicht beim TASC, weil dieser ganz andere String-Pointer als Applesoft benutzt. So steht beispielsweise das Länge-Byte vor dem String selbst und nicht beim Pointer wie in Applesoft. Es wäre deshalb erforderlich, das FRE-Programm völlig umzuschreiben. Übrigens, wenn man die RUNTIME von TASC 2.0 mit BLOAD RUNTIME, A$0806 einlädt, so befindet sich der TASC-eigene FRE-Befehl ab $0F16 im Speicher. us

RAMDISK.LC-Bug

Mein RAMDISK.LC-Programm enthielt leider einen unerfreulichen Bug. Es springt von der LC zunächst nach $0100 in den Stack, um dort die LC abzuschalten. Den Stack wählte ich, weil die unteren 48K, z.B. ab $0300 o.ä., unberührt bleiben sollten. Bei Applesoft- und Binärfiles war noch alles in Ordnung. Wenn jedoch *Zahlen*-Textfiles mit „FOR X = 1 TO N : PRINT Z (X): NEXT" gespeichert werden sollten, passierte folgendes: Jede einzelne Zahl Z (X) wird vom Applesoft-Interpreter in einen ASCII-String konvertiert, der zunächst ab $0100 abgelegt wird. Diese ASCII-Strings werden nach und nach in den I/O-Puffer z.B. ab $9600 übertragen. Wenn dieser Puffer voll ist, greift die RWTS und folglich auch der RAM-Disk-Driver ein, der den Bereich ab $0100 wegen des genannten Rücksprungs modifiziert. Und genau diese geänderten Bytes werden dann als Zahl auf den RAM-Disk-Sektor geschrieben, und zwar dann, wenn eine Zahl zwei Sektoren überlappt (z.B. die ersten 3 Ziffern am Ende des vorangehenden und die letzten 3 Ziffern am Anfang des nachfolgenden Sektors). Statt $0100 kann man z.B. $0110 als Rücksprung wählen, womit der Bug beseitigt wird. Es ist dann wie folgt zu patchen:

```
BLOAD RAMDISKLC
CALL -151
63CC: 10
63D4: 13
63D7: 10
63E9: 10
Ctrl-C
BSAVE RAMDISK.LC, A24861, L739
```

Die nachfolgenden Leserbriefe beziehen sich auf diesen Bug. Man beachte die geniale Lösung von Herrn Kociemba, die jedoch nur auf dem Apple II Plus und dem momentanen IIe funktioniert, nicht jedoch auf dem Apple IIc sowie auch nicht auf dem Apple IIe mit den neuen ROMs, weil sich $40 = Opcode für RTI bei $FFFE befinden muß:

Als Leser, der Ihre Zeitschrift von Anfang an regelmäßig liest, möchte ich nach der Mai-Ausgabe nun auch eine kleine Kritik anbringen.
Zum RAM-Disk-Driver-Artikel möchte ich fragen, warum der Umweg über Stackbereich oder eine 158,- DM teure Interface-Karte genommen werden muß. Es wäre doch theoretisch denkbar, die Routine zur Übertragung von „oben" nach „unten" parallel im Hauptspeicher und im Zusatzspeicher unterzubringen, so daß der Prozessor nach dem Umschalten die Routine im RAM auf der 64K-Karte weiter abarbeiten kann. Eine derartige Methode hätte zwar den Nachteil, daß etwas mehr Speicher für das Programm gebraucht wird, wäre aber eleganter als eine Zerstörung des Stacks oder der Gebrauch einer teuren Zusatzkarte. Ihre Ankündigung 4 weiterer Programme für diese Karte läßt mich einen Test dieser Karte fordern.
H.-D. Siewert, Lippstadt

Mit Interesse habe ich Ihren Artikel über die „Poor Man's RAM-Disk" gelesen. Sie schreiben, daß es nicht möglich ist, die Language Card von der Language Card aus zu desaktivieren. Mit einem kleinen Programmiertrick ist dies jedoch immer problemlos möglich, wobei das Programm jedoch von der Art des verwendeten ROMs abhängen kann.
Das hier beschriebene Programm funktioniert mit dem Autostart ROM, das wohl bei der überwiegenden Mehrzahl aller Apple verwendet wird, und beruht ausschließlich auf der Tatsache, daß der IRQ-Vektor dieses ROMs auf die Routine $FA40 zeigt. Dabei sind die Zeilen 471 bis 513 des von Ihnen beschriebenen RAM-Disk-Drivers durch folgende Zeilen zu ersetzen:

```
WEITER    EQU $BD06
EXIT1     JSR ZEROLOAD
          LDA /WEITER
          PHA
          LDA #WEITER
          PHA
          PHP
          JMP LCOFF
EXIT2     JSR ZEROLOAD
          TSX
          PHP
          INX
          INC STACK,X
          BNE LCOFF
          INX
          INC
          INC STACK,X
          JMP LCOFF
ZEROLOAD  LDX #3
ZEROL1    LDA ZERO,X
          STA RAM,X
          DEX
          BPL ZEROL1
          LDY ≤2
          RTS
          DFS $7
LCOFF     BIT $BIT $C082
```

Herbert Kociemba, Darmstadt

Es geht auch anders

Äußerst zuvorkommend war der Service der Firma Erphi-Elektronik, die nicht nur vollkommen kostenlos das EPROM meines Erphi-Controllers gegen einen neueren Stand austauschte, sondern auch gegen einen niedrigen Betrag mein eigenes 2 x 80 Track Philips-Laufwerk für Apple-Standard modifizierte.
Sowohl mit Laufwerk und speziell mit dem Controller bin ich mehr als zufrieden!
Leider gibt es auch Firmen mit anderer Geschäftsauffassung – besonders nach Verkauf der Produkte.
Es würde mich freuen, wenn dieser Brief veröffentlicht würde. Ich habe Hoffnung, daß derartiges Aufzeigen von sehr positiver Kundenbetreuung auch andere Hersteller/Händler zu ähnlichem Verhalten animiert.
Udo W.Völker, Essen

PLOT.2.0

Im Programm PLOT.2.0 in Peeker 5/85 auf Seite 17 haben sich leider zwei Fehler eingeschlichen:
3940 GOSUB 4700: IF A$ = "N" THEN 3910
5400 CALL AS%: F$ = "Y= ": GOSUB 2
Außerdem könnte man den EXEC-File weglassen, wenn man Zeile 5610 folgendermaßen ändert:
5610 POKE 103,1: POKE 16384,0: PRINT CHR$(4) "RUN PLOT.2.0".
Alois Strobel, Forbach
Vielen Dank für die Korrektur. „GOSUB 1010" geht „auf unsere Kappe", da wir u.a. die ersten Zeilen umgeschrieben haben, weil man sonst Zeile 2 nicht mehr eingeben könnte. us

DOS in LC

Obwohl mir Peeker sonst ganz gut gefällt, finde ich es nicht gut, daß angekündigte Beiträge nicht gedruckt werden, wie z.B. die Assembler-Einführung oder das 1-Drive-Kopierprogramm, und daß Serien sehr weit auseinandergezogen werden und nicht in aufeinanderfolgenden Ausgaben erscheinen (siehe Graf-quattro). Übrigens: Einen Artikel über das Verschieben des DOS in die LC finde ich wünschenswert.
Torsten Wolf, Bochum

SCREEN 80

Ich habe seit ein paar Wochen Zugang zu einem Apple IIc-System mit externer Floppy und Imagewriter. Weiterhin lese ich seit ein paar Monaten die Peeker-Hefte.
Nun verfüge ich zwar über Grundkenntnisse in BASIC, habe aber leider keine Ahnung und äußerst wenig Zeit, mir diese in Assembler-Programmierung zu verschaffen. Dennoch möchte ich aber die in Peeker erscheinenden Programme in eigenen Programmen als Utilities oder selbständig verwenden und einsetzen, da sie meist die Sparten abdecken, die ich als Nur-Anwender gut gebrauchen kann. Wie gebe ich aber die mit BASIC gemischten oder reinen Assembler-Listings und Hex-Dumps ein? – Kann ich die Programme von Peeker-Disketten ohne weiteres in eigene Programme einbinden?
Als Beispiel sei die Hard-Copy-Routine „SCREEN80" aus Peeker 4/85 gewählt. Ich tippte das Programm laut Listing ein – keine Reaktion nach Starten durch RUN oder CALL 768. – Wie muß ich vorgehen? – Wie kann ich feststellen, bis zu welcher Adresse mein BASIC-Programm geht und wo ich dann SCREEN80 anhängen oder vom BASIC-Programm aufrufen kann.
Michael Tarnowski, Wiesbaden
Wenn man SCREEN80 mit CALL

768 startet, muß der Drucker bereits eingeschaltet sein (grüne „POWER"-Taste leuchtet). Wenn der Drucker ausgeschaltet ist, wartet SCREEN80 „ewig", bis man ihn einschaltet. Dann hilft nur noch entweder Drucker einschalten oder Reset drücken. Da SCREEN80 für Drucker aller Art geschrieben wurde, kann die Druckbereitschaft nicht geprüft werden.us

Wie gibt man Maschinenprogramme ein?
Könnten Sie vielleicht einmal exakt erklären, wie man Programme (wie in Heft 5/85, S.50 INALL = Maschinenprogramm) in den Rechner eingibt?! Genaue Vorgehensweise beim Eintippen solcher Maschinenprogramme!
Roland Sieling, Voerde

AUGE und Peeker
Im Peeker, 4/85, S. 7, hatte ich geschrieben: „Die einschlägigen Treffen der AUGE usw. dienen meist weniger dem Erfahrungs- als vielmehr dem Programmaustausch, wobei die kopiergeschützten Programme mittels Kopien der kopiergeschützten Kopierprogramme kopiert werden". Diese Aussage beruht auf meiner Erfahrung mit einer Regionalgruppe, wo ich einmal vor einem guten Jahr 12 Apples mit „Nibble Away", „Locksmith" usw. in Aktion zählte, ferner auf telefonischen und schriftlichen Angaben von Peeker-Lesern und letztlich auch aufgrund von Mitteilungen im AUGE-Magazin selbst (zuletzt im Mai-Heft, S. 8: „Dann kam das Gespräch wieder auf die Regionalgruppen, wo es (Schilderung eines RG-Leiters) so aussieht: Mehrere Mitglieder bilden riesige Trauben um die 2-3 vorhandenen Computer, um kommerzielle Software zu kopieren").
Es wäre falsch zu sagen, daß *alle* Regionalgruppen sich diesbezüglich betätigen. Es wäre aber umgekehrt ebenso falsch zu sagen, daß *gar keine* Regionalgruppe sich diesbezüglich betätigt. Es gibt Regionalgruppen, die von vornherein das Kopieren von Fremdprogrammen untersagt und damit Schwarzkopierern einen Riegel vorgeschoben haben. Eine solche vorbildliche Regionalgruppe ist beispielsweise diejenige vom Großraum Kassel, wie aus dem nachfolgenden, aus Platzgründen gekürzten Leserbrief hervorgeht. Ich würde mich freuen, wenn alle Regionalgruppen so wären, und verzichte deshalb auch darauf, einen Leserbrief, der eine genau diametral entgegengesetzte Regionalgruppe schildert, abzudrucken, da ich mir von einer positiven Darstellung einen positiven Nachahmungseffekt verspreche. Zudem würde ich der AUGE empfehlen, in die Satzung einen Passus aufzunehmen, der Schwarzkopierer aus dem Verein ausschließt. Danach würde eine Selbstreinigung stattfinden und das Gerücht von einem „Kopierverein" ein für allemal aus der Welt geschafft. Nunmehr der Leserbrief:
In Ihrem Artikel über „Software und Kopierrecht", Heft 4/85, Seite 7, degradieren Sie, Herr Stiehl, die AUGE zu einem „Kopierverein". Ich bin seit Mitte 1981 Regionalleiter der AUGE im Großraum Kassel gewesen, und ich versichere hiermit, daß auf keinem einzigen von

Wie gibt man Maschinenprogramme ein?

Nicht jeder versteht sich in der Kunst der Maschinensprache. Die im Peeker veröffentlichten Assemblerprogramme sind dennoch auch für den nützlich, der nicht über die Kenntnis von Bits und Bytes im Computer verfügt. Wir wollen daher im folgenden erläutern, wie Maschinenprogramme eingegeben, gespeichert und aufgerufen werden.

Die Eingabe
Zur Eingabe von Hexadezimal-Zahlen bedient man sich am besten des System-Monitors, der aus Applesoft durch **CALL -151** aktiviert wird. Es erscheint dann anstatt des gewohnten Promptzeichens „Ü" oder „]" der Stern „*". Will man nun einen Hexdump eingeben, so tippt man zunächst die erste Adresse *ohne* „$", gefolgt von einem Doppelpunkt (alle Werte werden im Monitor stets hexadezimal angegeben). Die folgenden Hex-Zahlen werden nun, jeweils durch ein Leerzeichen getrennt, nacheinander eingegeben (*kein* Leerzeichen zwischen Doppelpunkt und erster Zahl), wobei *nicht* nach jeweils 8 Bytes die Adresse neu eingegeben werden muß (wie im Hexdump ausgedruckt), sondern bis zu ca. 80 Hex-Zahlen getippt werden können, bevor die Return-Taste betätigt werden muß. Um weitere Zahlen einzugeben, genügt es nun, wieder einen Doppelpunkt einzugeben und die nächste Zahlenkolonne einzutippen. So verfährt man, bis alle Bytes eingegeben sind.

Assemblerlistings
Wurde nur ein Assemblerlisting abgedruckt, so verfährt man hierbei analog, wobei nun alle diejenigen Bytes eingegeben werden müssen, die rechts von dem Doppelpunkt der Adresse und der fortlaufenden Zeilennummer des Listings stehen. Das kurze Assemblerlisting

```
0300:  A9 B0      1 LDA #"0"
0302:  20 ED FD   2 JSR COUT
```

wird durch folgende Zeile eingegeben:

300:A9 B0 20 ED FD <RET>.

Nachprüfen und Korrigieren
Zuweilen möchte man die eingegebenen Zahlen nachprüfen. Dazu gibt man die Anfangsadresse des gewünschten Bereichs, einen Punkt und dann die Endadresse des zu listenden Bereichs ein, gefolgt von Return (z.B. „300.345 <RET>"). Es erscheint dann eine Liste der gespeicherten Bytes, wie sie in den Hexdumps ausgedruckt ist.
Hat man einen Fehler entdeckt, so kann ein einzelnes Byte durch „<Adresse>:<Byte>" überschrieben werden. *Achtung:* Steht das Byte nicht am linken Rand, so muß die tatsächliche Adresse vom Rand her abgezählt werden. Ist z.B. in der Zeile

0308- 00 01 03 03 04 05 06 07

das dritte Byte falsch, so wird es durch

30A:02 <RET>

überschrieben. Durch Eingabe von

308.30F <RET>

wird dann die korrigierte Zeile angezeigt:

0308- 00 01 02 03 04 05 06 07.

Hat man ein Byte zuviel oder zuwenig eingegeben, hilft fast nur noch ein erneutes Eintippen ab der fehlerhaften Stelle.

Abspeichern
Bevor nun der erste Probelauf erfolgt, sollte das Programm zunächst abgespeichert werden. Hierzu tippt man die Kopfzeile des Hexdumps ein (z. B. „BSAVE PROGRAMM, A$300, L$AB <RET>") oder allgemein „**BSAVE** <Name>, A$<Start>, L$<Länge>". Dabei ist <Start> die erste Adresse des Programms (hexadezimal) und <Länge> die Anzahl von Speicherstellen, die es belegt. Die Länge kann bei Listings der letzten Zeile entnommen werden. „171 Bytes" steht z.B. für *dezimal* 171 Bytes (hexadezimal $AB Bytes); soll der dezimale Wert angegeben werden, so muß das Dollarzeichen bei der Länge-Angabe entfallen: „BSAVE PROGRAMM, A$300, L171 <RET>".

Laden und Aufrufen
Zum späteren Einladen des Programms genügt die Eingabe „**BLOAD** <Name>" oder, falls es sofort ausgeführt werden soll, „**BRUN** <Name>". Ein im Speicher befindliches Programm startet man im Monitor durch „<Start> G", wobei <Start> wieder für die hexadezimale Startadresse des Programms steht.
Hat man seine Arbeit im Monitor beendet, so kommt man durch Eingabe von Ctrl-C <RET> wieder nach Applesoft.

mir geleiteten Treffen jemals andere als selbsterstellte oder freigegebene Vereinssoftware kopiert worden ist! Meine Regionalmitglieder können durchweg bezeugen, daß ich immer energisch gegen alle anderen Kopierversuche vorgegangen bin und daß niemals der Einsatz eines Bit-Kopierers o.ä. geduldet worden ist! Unsere Treffen hatten immer Niveau und dienten – abgesehen von ungeliebtem, aber notwendigem Vereinskram – ausschließlich der Information und Weiterbildung, der Weitergabe von Erfahrungen, der Lösung von Problemen, die eigentlich Angelegenheit der Firma Apple gewesen wäre, und der Unterstützung von Neulingen. Aus unserer Gruppe entstand irgendwann dann die Regionalgruppe Göttingen, der zum größten Teil auch Lehrpersonal der Universität angehört, und die sich, wie ich mir vorstellen kann, sicherlich ebenso wie wir gegen Ihre unqualifizierte globale Behauptung verwahren wird.

Wenn man allerdings die Seite 68 desselben Heftes aufschlägt, in dem Sie regelrechte Rundumschläge in Richtung AUGE austeilen, wird einiges klar: Sie selbst sind ja ebenfalls AUGE-Mitglied, und zwar ein ebenso engagiertes wie ich. Einen Beweis dafür kann ich vorlegen in Form einer erstklassigen „AUGE-Diskette" über das Grundthema „Assembler-Programmierung", die Sie verfaßt haben und die in unserer Regionalgruppe immer noch sehr geschätzt wird. So fleißige und engagierte Leute wie Sie stoßen in einer Vereinigung von inzwischen fast 5000 Mitgliedern allerdings zwangsläufig irgendwann einmal auf Frust, und Sie hatten offensichtlich Ihr größtes Negativ-Erlebnis mit einem rudimentären, schwachen und ziemlich kleinkarierten Vorstand der AUGE, den Sie nun gleichsetzen mit dem gesamten Verein! Sie bringen unsere wirklich interessierten und guten Mitglieder durch Ihre Verhaltensweise in ein arges Dilemma: Denn einerseits schätzen wir Ihre Zeitschrift vom sonstigen fachlichen Inhalt her als eine der zur Zeit besten auf dem Markt. Dabei identifizieren wir uns zu einem Großteil insbesondere mit der von Ihnen vertretenen kritischen Haltung gegenüber dem Geschäftsgebaren der Firma Apple, von der wir als Einzelpersonen, aber auch als AUGE-Mitglieder in der Vergangenheit wahrhaftig nicht verwöhnt worden sind. Andererseits können wir nicht Peeker-Abonnenten und AUGE-Mitglieder gleichzeitig sein, wenn Sie eine Ihrer wichtigsten Aufgaben darin sehen, gegen uns Krieg zu führen! Auch glauben wir, daß ja das „User Magazin" und der „Peeker" nicht um jeden Preis in „Konkurrenz" zueinander stehen müssen.

Jürgen Heil, Volkmarsen

An Ear to the Ground

Nach jedem Peeker-Heft gehen so viele Zuschriften ein, daß ich pro Monat eine ganze Woche nur mit Leserkorrespondenz befaßt bin. Es ist mir ein besonderes Bedürfnis, mich an dieser Stelle herzlich für dieses „Feedback" zu bedanken. Grundsätzlich wird jedes Schreiben von mir beantwortet, doch mögen Sie bitte Verständnis dafür haben, daß es etwa 3 Wochen dauern kann, bis Sie eine Antwort erhalten. Aus Platzgründen kann ich nur einen Bruchteil der Zuschriften in Leserbriefe für den Peeker ummünzen.

Für den Fall, daß es Sie interessiert, warum ein im April-Heft des AUGE-Magazins veröffentlichter Pro-Mac-Contra-Stiehl-Leserbrief nicht im Peeker erschien, kann ich nur kryptisch antworten, daß mir vor einiger Zeit eine gewisse Computerfirma unter Zeugen die Kopie eines an den Peeker gerichteten gewissen Pro-Mac-Leserbriefes zeigte. Seit dieser Zeit reagiere ich auf Pro-Mac-Leserbriefe allergisch, denn Sie könnten ja mit einer gewissen Computerfirma besprochen worden sein. Dies ist doch wohl verständlich, oder nicht?

U. Stiehl

Extended Graphic System

getestet von Dr. Jürgen B. Kehrel

Ohne Hilfsmittel ist die Programmierung von Apple-Shapes eine sehr mühsame und fehleranfällige Beschäftigung. Hier hilft Ihnen das Extended Graphic System (DM 198,-, Firma Interface Age), das trotz seines Namens aus Deutschland kommt. Es dient zum Erzeugen und Editieren der hochauflösenden Grafiken (HGR) sowie dem Erstellen von Shapes und den aus diesen bestehenden Tabellen. Sie können im vergrößerten Maßstab Ihre Figuren mit den Cursortasten oder einem Joystick entwerfen, drehen, in der Farbe variieren, ganz so wie es Ihnen gefällt, aber auch in der Originalgröße überprüfen. Ihr Produkt läßt sich auf Diskette sichern oder mit anderen zusammen in eine Vektorentabelle einbinden. Auch aus fremden Tabellen können Sie einzelne Shapes isolieren und abspeichern. Extended Graphic System sorgt für die Einhaltung aller Regeln und liefert ohne weiteres Zutun komplette Applesoft-Vektorentabellen, auf die Sie dann aus Ihren eigenen Basic- oder Maschinenprogrammen zugreifen können. Ein Demonstrationsprogramm führt einige Beispiele vor. Da alle Teile menügesteuert ablaufen, werden Sie die gut 40 Seiten der Bedienungsanleitung (mit übermäßig vielen Zeichensetzungsfehlern) nicht häufig zu Rate ziehen müssen. Shapes aus beliebig vielen Tabellen können nacheinander geladen, manipuliert sowie zu einem Bild kombiniert werden. Ihr „Werk" läßt sich in 5 Größenstufen mit wählbarem Rand, auch mehrfach neben- oder untereinander ausdrucken. Das Dump-Programm ist abgestimmt auf EPSON-Drucker der 80er-Serie, ITOH 1550/8510A sowie kompatible Geräte.

... für APPLE und APPLE-compatible ...

- 16 K Karte
- Z 80 Karte
- Centronics Drucker Interface (grafikfähig)
- Musik-System (3 stimm. Synthesizer)

für je **DM 85,–**

Alle Karten 100 % getestet – volle Garantie.
Sofort ordern im

COMPUTERLADEN MEMMINGEN
Kreuzstraße 9 · 8940 Memmingen · Telefon 08331-5942

Terminal-Programm für die Super-Serial-Card

von Andreas Lecreux

Das Terminal-Programm aus dem Peeker 4/85 wurde auf einer ASIO-kompatiblen Karte getestet und entwickelt. Da diese Karte nicht sehr verbreitet ist, folgt hier die Anpassung an die Super-Serial-Card von Apple. Die benutzten Adressen beziehen sich auf diese Karte in Slot 2 und sind dem dazugehörigen Handbuch (Apple #A2L0044) entnommen.
Die Programme werden erst vor der Ausführung geändert, damit das Original erhalten bleibt. Dies wird durch einen EXEC-File erreicht, der als erstes die Originalprogramme in den Speicher lädt, diese dann entsprechend patcht und danach ausführt. Der Aufruf erfolgt somit durch „EXEC **TERMINAL.PATCH**".
Bei jedem dieser Schritte wird auf dem Bildschirm eine Meldung ausgegeben. Wer diese Prozedur nicht jedesmal ausführen will, kann die geänderten, im Speicher befindlichen Programme auf einer anderen Diskette durch „SAVE TERMINAL" und „BSAVE TERMINAL.B, A$2000, L1046" abspeichern.
Im Applesoft-Programm werden die Zeilen, in denen die Geschwindigkeit (Zeilen 2100 ff.) und die Steuerleitung RTS gesetzt werden, ersetzt. Die genauen Änderungen entnehme man dem Listing. In der Programmzeile 2904 wird die Schnittstelle initialisiert. Die hier verwendeten Parameter bedeuten:
0C – keine Verzögerung nach einem CR,
1D – 7 Datenbits und 1 Stoppbit
0P – ohne Parität,
1T – keine Übersetzung von Kleinbuchstaben,
80N – 80 Zeichen/Zeile,
0L – keine Verzögerung nach einem LF.
Als zweites erfolgt ein Sprung in den Monitor. Von hier aus werden im Maschinenprogramm alle Adressen, bei denen auf die Status- und Datenregister der Schnittstelle zugegriffen wird, gepatcht. Nach dem Rücksprung zu Applesoft wird das neue Programm gestartet.

TERMINAL.PATCH

```
NOMON C,I,O
HOME : PRINT "Patch für Super-Serial-Card"
PRINT "nur bei Version 1.3"
PRINT "BASIC-Programm wird geladen"
LOAD TERMINAL
PRINT "Maschinenprogramm wird geladen"
BLOAD TERMINAL.B
PRINT "Patch der Programme"
PRINT "     1. Geschwindigkeit"
DEL 9014,9034
1590
1535 DIM SP(15)
1536 RESTORE : FOR I = 1 TO 15
1537 READ SP(I) : NEXT
1700 DATA 50,75,110,135,150,300,600,1200
1710 DATA 1800,2400,3600,4800,7200,9600
1720 DATA 19200
2105 PRINT "Tempo zur Zeit = ";SP(SP)
2107 FOR K = 1 TO 15 : PRINT K;" = ";SP(K);
2110 PRINT " Baud" : NEXT K
2115 INPUT "";SP$ : SP = VAL(SP$)
2120 IF SP < 1 OR SP > 15 GOTO 2115
2125 PRINT D$"PR#"SE : PRINT I$;SP;"B" : PRINT D$"PR#"SC
2904 PRINT D$"PR#"SE : PRINT I$"0C1D0P1T80N0L" :
     PRINT D$"PR#"SC
PRINT "     2. RTS setzen und löschen"
2805 :
3330 :
2602 :
3050 :
3055 :
PRINT "     3. Ändern des Maschinenprogrammes"
CALL -151
2081:A8 N 22EF:A8 N 232D:A8 N 2354:A8
23A9:A8 N 2079:A9 N 22E7:A9 N 2320:A9
234C:A9 N 23A1:A9 N 23EF:08 N 23F0:10
3D0G
2415 PRINT "Patch Super-Serial-Card beendet"
RUN
```

Diversi-DOS 4-C

getestet von Rudolf Rötering

Die vier bekanntesten Betriebssysteme für den Apple II sind:

– DOS: Disk Operating System,
– UCSD-Pascal: (zu Ehren des Philosophen Blaise Pascal),
– CP/M: Control Program for Microprocessor,
– ProDOS: Professional Disk Operating System.

Seit geraumer Zeit gibt es darüber hinaus eine Vielzahl von Disketten-Betriebssystemen, die auf DOS 3.3 basieren und im wesentlichen Geschwindigkeitsverbesserungen und teilweise mehr Befehle bieten. Diese applefremden Produkte kamen auf den Markt, weil die Fa. Apple die Programmpflege des betagten DOS-3.3-Systems (Aug. 1980) schlichtweg verschlafen hatte. Als man bei Apple endlich aufwachte, war es für eine verbesserte Version zu spät, da bereits applefremde DOS-Varianten vorlagen, die Apple selbst wohl kaum mehr hätte übertreffen können. Daraufhin wurde kurzerhand das bereits für den Apple III vorliegende SOS (Sophisticated Operating System) für den Apple II umgeschrieben, womit das mit DOS 3.3 wenig kompatible ProDOS entstand. Es wurde gegenüber DOS 3.3 so stark abgeändert, daß die alten Compiler (z.B. Tasc, der selbst DOS-neutral ist) nicht mehr einsetzbar sind.

Jedes System hat seine eigenen Vorzüge. Systeme mit geringen Änderungen haben den Vorteil der weitgehenden Kompatibilität mit DOS 3.3, bieten aber meist nur Geschwindigkeitsvorteile bei Ladevorgängen von Programmen. Das einzige Betriebssystem, das bei allen Schreib- und Leseoperationen Verbesserungen bietet – auch bei den besonders zeitkritischen Textdateien – ist das Diversi-DOS, zuletzt erschienen in der Version 4-C. (Die Version 4-C zeigt nach unseren Messungen keine Geschwindigkeitsverbesserung gegenüber der alten 2-C-Version.)

Die gelieferte Utility-Diskette überträgt das neue Betriebssystem menügesteuert auf DOS-3.3-Disketten. Dabei können aus diesem Menü heraus eine Vielzahl weiterer Optionen gewählt werden, wie z.B. die Ausgabe der Bedienungsanleitung auf Drucker oder als Textdatei.
Der wohl wichtigste Vorzug von Diversi-DOS ist die Geschwindigkeit beim Diskettenzugriff. **Tabelle 1** gibt einen Zeitvergleich zwischen ProDOS, DOS 3.3 und Diversi-DOS wieder.
Die höhere Geschwindigkeit von Diversi-DOS liegt nicht in einer schnelleren RWTS-Routine, sondern in der Verbesserung des sog. File-Managers. Um einen Sektor zu beschreiben, sind bei DOS 3.3 18 Umdrehungen der Diskette erforderlich, während Diversi-DOS nur 2 benötigt.
Der krasseste Unterschied liegt bei „APPEND" vor. DOS 3.3 liest den File so lange Feld für Feld ein, bis der Endmarker CHR$(0) erreicht ist. Diversi-DOS dagegen sucht den Sektor für das Ende der Datei in der TSL und liest ihn ein, um dort die nächsten Felder anzuschließen.

Tastatur- und Druckerpuffer

Diversi-DOS ermöglicht die Einrichtung eines Tastatur- und/oder Druckerpuffers in der LC oder anderen RAM-Karten, die nach Ihren Wünschen generiert werden können (nur beim 48K-DOS). Diese Option BUFFER schließt den Befehl „INIT" aus und verändert die Vektoren für die Ein- und Ausgaberoutinen des Monitors ($0036 bis $0039). Damit compilierte Programme, die ihrerseits auch die CHRGET-Routine verändern, lauffähig bleiben, kann z.B. durch „POKE 46965, 16" der CHRGET-Patch von BUFFER abgestellt werden. Ebenso sind alle Steuerbefehle an das Drucker-Interface ohne Funktion auf das Druckergebnis. Ob ein Videoecho erfolgt oder ob LF nach CR ausgeführt werden soll, muß mit POKEs in BUFFER gewählt werden. Der Vollständigkeit halber sei noch erwähnt, daß BUFFER mit Einschränkungen auch unter Integer-BASIC verwendbar ist.

Der FREE-Sector-Patch

Wenn Sie den FREE-Sector-Patch gewählt haben, kann während des Programmablaufes der freie Speicherplatz auf der Diskette ermittelt werden. Durch folgendes Beispiel wird dies verdeutlicht:

```
100 POKE 48376,96: CALL
    48351: POKE 48376,32
```

110 FR = PEEK(70) * 256 + PEEK(68) : REM FR = unbenutzte Sektoren

Für 64K-DOS muß Zeile 100 so aussehen:

100 POKE 63261,96: POKE 49058,0: CALL 49054: POKE 49058,181: POKE 63261,32

Es sollte natürlich kurz vorher ein Diskettenzugriff (z.B. ein CATALOG) erfolgt sein, damit der Wert auch aktuell ist. Eine mögliche Anwendung wäre die Überprüfung der Rest-Speicherkapazität einer Datendiskette bei der Buchführung, bevor die nächste Konten- bzw. Buchungsdatei eröffnet wird, damit auch alle Daten auf dieser Diskette Platz haben.

DOS in der Language-Card

Das 48K-DOS läßt sich mit „BRUN DDMOVER" in die Language-Card verschieben. Dadurch ist es zu einem 64K-DOS geworden. Aus einer ganz normalen DOS-3.3-formatierten Diskette können Sie mit der Utility-Diskette eine 64K-Version herstellen, die während des Bootens das DOS in die LC schiebt. Wenn Sie beim Booten die ESC-Taste drücken, kann der „MOVE" unterdrückt werden. Ist PEEK (55) > 190, wurde DOS in die LC geschoben. Das 64K-DOS stellt einige neue Befehle bereit:

- Beim **CATALOG** wird die Rest-Speicherkapazität der Diskette angezeigt.
- Mit **INIT X** lassen sich Datendisketten formatieren, die die Spuren, auf denen sonst das DOS gespeichert wird, für Daten freigibt. Dadurch erhält man 32 Sektoren = 8K mehr nutzbaren Speicherplatz.
- **HIMEM** liegt bei $BF00 (48896), wenn MAXFILES = 3 ist. Das bedeutet etwa 10K mehr RAM für Daten und Programme.
- Die **Fehlermeldungen** werden im Gegensatz zum 48K-DOS ausgeschrieben, und nicht nur mit Code-Ziffern bezeichnet.
- Mit **ESC** kann die CATALOG-Ausgabe nach jeder Bildschirmseite abgebrochen werden.
- Bei **BSAVE** wird an den Programm-Namen automatisch die letzte Binäradresse und Programmlänge angefügt. Mit **PAD** lassen sich diese Angaben auf den Bildschirm bringen.
- Der RUN-Befehl kann einen **„L"-Parameter** (Line) enthalten, der das Programm ab der angegebenen Zeile startet.
- Ein sehr nützliches Feature ist das **Wildcard-Zeichen** in Programmnamen. Mit einem „=", angehängt an die ersten Zeichen des Programm-Namens, wird der erste File im CATALOG gewählt, bei dem die Zeichen vor dem „=" zutreffen. Wenn anstelle des Gleichheitszeichens ein Fragezeichen eingegeben wird, muß mit „Y" oder „N" das Programm ausgewählt werden. Das Wildcard-Zeichen allein ist nicht möglich.
- Alle DOS-Befehle können auch in **Kleinschrift** eingegeben werden.
- Mit **„C"** erhalten Sie einen CATALOG vom zuletzt eingestellten Slot und Drive. Parameter sind hier nicht möglich.
- Mit **TLIST, <Name>** lassen sich z.B. Textfiles auf dem Bildschirm anzeigen. Mit der Leertaste hält die Darstellung an und fährt bei erneuter Betätigung fort.

Wenn ein BASIC-Programm geTLISTet wird, kann ein „L"-Parameter angefügt werden, der das Listing ab der angegebenen Zeile beginnen läßt. Hier bewirkt die Leertaste einen Wechsel zwischen Pause und dem Listing der nächsten Zeile. Jede andere Taste läßt den Listvorgang wieder kontinuierlich durchlaufen. Dabei kann mit „D" zwischen den Zeilen eine Leerzeile und mit „S" diese Option wieder abgestellt werden. Mit dem „:" kann das Listing, z. B. zur Dokumentation, sehr übersichtlich gestaltet werden: Bei Mehrfach-Statements werden die Befehlssegmente zwischen den Doppelpunkten untereinander gelistet. Das Semikolon „;" stellt diese Funktion wieder ab. Wenn Sie beim Programm-Namen ein BASIC-Programm wählen, können auch andere Programme mit allen möglichen Features geTLISTet werden, ohne das aktuelle Programm zu löschen. Hier können durch Überfahren mit dem Cursor auch Zeilen aus anderen Programmen übernommen werden. Während des TLISTings werden die Leerzeichen des Interpreters links und rechts der Zeile nicht gedruckt. Dadurch läßt sich beim Editieren der Cursor ohne Ausblendung bis zum Ende der Zeile fahren. Ctrl-Zeichen werden invers dargestellt und können beim Editieren gelesen werden.

- Während mit den Pfeiltasten editiert wird, können nach einem Ctrl-I neue Zeichen in die bestehende Zeile eingefügt werden. Den **INSERT-Modus** erkennen Sie daran, daß der Cursor nicht mehr blinkt und der Text links davon eingefügt wird, während der Cursor selbst seine Position beibehält. In diesem Modus können mit der Linkspfeiltaste die Zeichen wieder gelöscht werden. Durch ein erneutes Ctrl-I wird der INSERT-Modus wieder verlassen.
- Die wohl leistungsfähigste Option beim 64K-DOS sind die frei definierbaren **Tastatur-MACROS**. Jede der Tasten kann mit einer Folge von Zeichen definiert werden, von denen aber nur die Ctrl-Sequenzen verwendet werden sollten. Auf diese Weise können Sie sich für die verschiedensten Anwendungsfälle Macros definieren und in einem Binärfile mit frei wählbarem Namen z. B. mit „BSAVE MACRO.E, A$DC00, L$0100" abspeichern.

RAM-Disk

Vom gleichen Hersteller kann ein RAM-Disk-Emulator bezogen werden, für den Sie einen 64K-Apple mit zusätzlich mindestens 64K RAM und Diversi-DOS in der Language-Card benötigen. Diese Voraussetzungen erfüllen ein Apple IIc, ein Apple IIe mit 64K Aux-Memory oder ein 64K Apple II Plus mit 2 zusätzlich möglichen 128K-RAM-Karten in Slot 1–7. Für die 128K-Typen IIc und IIe ist der vorhandene RAM-Disk-Driver schon automatisch konfiguriert. Sollten Sie von dieser Konfiguration abweichen, muß der RAM-Disk-Driver neu eingestellt werden. Zusätzlich zur normalen Rechnerkombination kann noch eine 2. Karte in die RAM-Disk einbezogen werden, was mit dem mitgelieferten Programm RDCONFIG erfolgt. Angesprochen wird die RAM-Disk über den Pseudo-Slot 3 mit „,S3". Die Laufwerksangabe kann entfallen.

Als raffinierte Option ist hier die Möglichkeit zu nennen, eine Diskette so zu initialisieren, daß ein begrenzt formatierter Inhalt in die RAM-Disk paßt. Bei diesem Vorgang wird auch der Driver und das Startprogramm RDHELLO dort gespeichert. Danach ist das Betriebssystem auf 64K-DOS umzustellen. Beim Booten dieser Diskette wird die RAM-Disk initialisiert und der Disketteninhalt dorthin kopiert. Diese Option erspart das mühsame Kopieren von Textfiles in die RAM-Disk mit READ und WRITE oder mit FID. Eine solche Bootdiskette muß nicht, sollte aber vorher auf allen Tracks formatiert werden, weil ein Kopieren mit COPYA sonst nicht möglich wäre.

Die Anzahl der formatierten Tracks hängt von der Einstellung des Drivers ab, der wiederum an die RAM-Karten angepaßt werden muß. Dieses erledigt ein auf der Utility-Diskette befindliches Programm namens RDCONFIG. Leider werden die Optionen hinsichtlich reservierter Bereiche für Double-HIRES, Konfigurationsmeldung beim Booten und Bank-2-Schutz für die Karten im Aux-Slot 3 nicht berücksichtigt.

Bei der Installierung der RAM-Disk wird HIMEM um 768 Bytes auf $BC00 (48128) herabgesetzt, um Platz für den Driver zu schaffen. Hiernach stehen noch satte 45K RAM für Daten und Programme zur Verfügung. Auch die Bank 1 der Language-Card wird automatisch in die RAM-Disk integriert, die jedoch durch die Driver-Einstellung ausgeschlossen werden kann, um z.B. die schnelle Garbage-Collection von Harald Grumser (Peeker, Heft 1/2, 85) benutzen zu können, die ja bekanntlich die Bank 1 belegt.

Die RAM-Disk-Einstellung wird durch die in **Tabelle 2** aufgeführten Speicherstellen bestimmt.
Tabelle 3 zeigt eine Übersicht über die freien Speicherkapazitäten von Bootdisketten und RAM-Disks im Aux-Slot 3 des Apple IIe mit der INIT-Option des RAM-Disk Drivers.

Bootdisketten für 128K-Karten stellen 480 und darüber hinausgehende Karten z.B. die 192K-Neptune 496 freie Sektoren (volles DOS-3.3-Format) zur Verfügung. Wie weiter oben schon erwähnt, läßt sich zu einer dieser Karten in Slot 1–7 eine weitere RAM-Karte in die RAM-Disk einbinden.
Wenn der RAM-Disk-File im Immediate-Mode geBRUNt wird, stehen dort 6 Sektoren mehr zur Verfügung.

Zusammenfassend läßt sich sagen, daß ein Apple IIe mit 64K Aux-Memory oder größer, LC.FREE-Garbage-Collection in Bank 1 der Language-Card, 64K Diversi-DOS mit RAM-Disk und 45K freiem RAM eine optimale Nutzung ermöglicht.

Tabelle 1

Zeile	Befehl	ProDOS	Diversi-DOS 4-C	DOS 3.3
10	BSAVE	383 sec	155 sec	837
20	BLOAD	97 sec	125 sec	628
30	WRITE	30 sec	57 sec	140
40	READ	27 sec	47 sec	165
50	WRITE	108 sec	95 sec	171
60	READ	105 sec	86 sec	243

Tabelle 2

Option bei 64K-Karten in Slot #3	Bootdisk		RAM-Disk	
	Sekt.	KByte	Sekt.	KByte
ohne	224	56	244	61
Double:				
HIRES 1	176	44	196	49
HIRES 2	144	36	164	41
HIRES 1 + 2	144	36	164	41
ohne Bank 2	wie mit		– 8	– 2

Anmerkungen zu Tabelle 1

Die Werte wurden mit dem internen Laufwerk des Apple IIc ermittelt. Die in der Tabelle angegebenen Zeilennummern beziehen sich auf das folgende Programm:

```
10 FOR I = 1 TO 20: PRINT CHR$ (4)"BSAVE XXX, A$1000,
   L$7FFF": NEXT : END

20 FOR I = 1 TO 20: PRINT CHR$ (4)"BLOAD XXX": NEXT : END

30 A$ = "123456789 123456789 123456789 123456789 123456789
   123456789 123456789 123456789 123456789 123456789"
31 PRINT CHR$ (4)"OPEN YYY": PRINT CHR$ (4)"WRITE YYY":
   FOR I = 1 TO 500
32 PRINT A$
33 NEXT : PRINT CHR$ (4)"CLOSE": END

40 PRINT CHR$ (4)"OPEN YYY": PRINT CHR$ (4)"READ YYY":
   FOR I = 1 TO 500
41 INPUT "";A$
42 NEXT : PRINT CHR$ (4)"CLOSE": END

50 A = 123456789
51 PRINT CHR$ (4)"OPEN ZZZ": PRINT CHR$ (4)"WRITE ZZZ":
   FOR I = 1 TO 5000
52 PRINT A
53 NEXT : PRINT CHR$ (4)"CLOSE": END

60 PRINT CHR$ (4)"OPEN ZZZ": PRINT CHR$ (4)"READ ZZZ":
   FOR I = 1 TO 5000
61 INPUT "";A
62 NEXT : PRINT CHR$ (4)"CLOSE": END
```

Tabelle 2

```
(8201) $2009:  1. RAM-Karte

          0 =  64K IIe RAM-Karte     (Aux)
          1 = 192K Neptune IIe       (Aux)
          2 = 128K Memorymaster IIe (Aux)
    Slot*16 = 128K Saturn Typ  (Slot 1-7)
128+Slot*16 = 128K Legend Typ  (Slot 1-7)

(8202) $200A:  2. RAM-Karte

    Slot*16 = Saturn Typ (Slot 1-7)
128+Slot*16 = Legend Typ (Slot 1-7)

(8203) $200B: Konfigurationsmeldung
              bei RAMDISK unterdrücken

              0 = ausgeben
              1 = unterdrücken

(8204) $200C: Initialisieren der RAMDISK
              beim Booten mit Kopie

              0 = nein, 1 = ja

(8205) $200D: Bank 2 schützen

              0 = nein, 1 = ja

(8206) $200E: Double-Hires-Schutz

              0 = kein Schutz
              1 = Double-Hires 1
              2 = Double-Hires 2
              3 = Double-Hires 1+2
```

Diversi-DOS 2-C auf Peeker-Sammeldisk #6

In Absprache mit der Firma Diversified Software Research haben wir auf die Sammeldisk #6 zwar nicht das ganze Betriebssystem, wohl aber die Dateien HELLO und ASMDIV aufgenommen. Wenn Sie Diversi-DOS benutzen wollen, bitten wir Sie, $30.00 (30,- Dollar) in Form eines auf eine beliebige amerikanische Bank ausgestellten Schecks (ggf. Kredit-Karte) zugunsten von DSR an folgende Anschrift zu senden:

DSR, Inc.
5848 Cramptom Ct.
Rockford, IL 61111
USA

Konto-Nr.815 877-1343
(Visa/Master-Card)

Sie erhalten dann direkt von DSR die Diskette mit der neuesten Version 4-C und allen dazugehörigen Utilities.
Zur Installierung der auf der Sammeldisk enthalten Version 2-C verfahren Sie wie folgt:

1. DOS 3.3 von System-Master-Diskette booten
2. NEW
3. 10 PRINT "HELLO"
4. Leerdiskette einlegen
5. INIT HELLO
6. Peeker-Sammeldisk einlegen
7. RUN HELLO
8. Formatierte Leerdiskette einlegen

Nunmehr aus HELLO-Menü nacheinander folgende Optionen (=Ziffer) wählen:

Option 2 = PUT DIVERSI-DOS ONTO A DISK
Option 7 = BSAVE DDMOVER FILE
Option 9 = EXIT TO BASIC

Damit ist aus Ihrer formatierten Leerdiskette eine bootfähige Diversi-DOS-Diskette geworden. Diversi-DOS 2-C unterscheidet sich von 4-C in folgenden Punkten:

1. 2-C enthält außer dem DDMOVER keine zusätzlichen Features („Schnickschnack"), so daß die 48K-Version viele Einsprungadressen vom alten DOS 3.3 bewahrt. Persönlich benutze ich deshalb nur 2-C.

2. 2-C ist genauso schnelle wie 4-C.
3. Wenn man mit BRUN DDMOVER 2-C in die Language-Card Bank 2 schiebt, so liegt HIMEM fest bei $BF00 (48896). MAXFILES ist auf 5 festgelegt, d.h. 5 OPEN-Files. Bei 4-C wird bei Bedarf der I/O-Puffer ab $BF00 abwärts fortgesetzt. Bei 2-C kann man also HIMEM herabsetzen und Maschinenprogramme bequem oberhalb von HIMEM unterbringen. Dies ist bei 4-C nicht ohne weiteres möglich.
4. Die diversen RAM-Disk-Driver für die 64K-Karte auf der „MMU-2.0-Diskette" (Hüthig Software Service) sowie aus „Apple DOS 3.3, Tips und Tricks" laufen nur in Verbindung mit 2-C.

Nachdem Sie die Originalversion 4-C von Diversified Software Research erworben haben, können Sie selbst wählen, welche Version Ihnen mehr zusagt.

Lasar II ze

getestet von H. Grumser

Die Firma MICROMINT Streil bietet zu einem Grundpreis von DM 1365,- einen zum Apple II Plus kompatiblen Rechner an, der bereits über eine 16K-Speichererweiterung und den Z80-A-Prozessor auf der Mutterplatine verfügt.
Der Lasar II ze ist in einem etwas üppigen IBM-ähnlichen Gehäuse untergebracht, das zwei normale oder vier Slimline-Laufwerke aufnehmen kann. Als externe Tastatur wird ein MARK-II-Keyboard der Firma Multitech geliefert.
Die zum Test bereitgestellte Ausstattung umfaßte außerdem den Monitor Lasar 4000 (DM 330,-), ein bereits im letzten Peeker getestetes Chinon-Laufwerk mit Controller (DM 645,-) und eine 80-Zeichenkarte (DM 179,-).

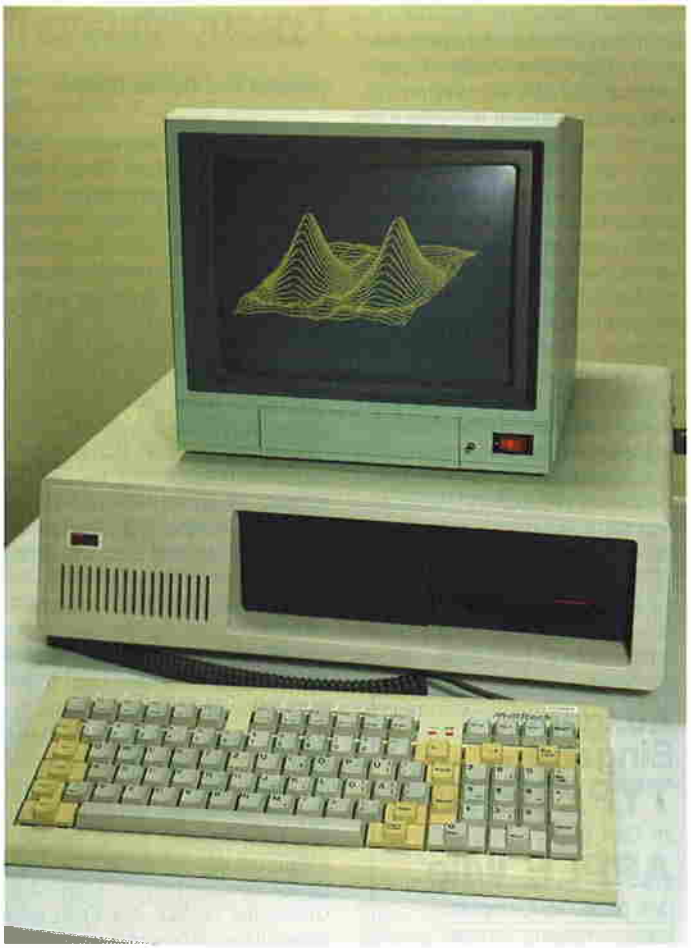
Aufbau und Inbetriebnahme

Die erste zu nehmende Hürde ist der Zusammenbau der einzelnen Komponenten, die ordentlich verpackt geliefert werden. Die Beschreibungen der einzelnen Teile ist, abgesehen vom Handbuch der Tastatur, etwas dürftig, was dem Apple-Neuling Schwierigkeiten bereiten dürfte. Mit etwas technischer Intuition ist der Aufbau jedoch zu handhaben und nach anfänglicher Mühe erscheint zum ersten Mal die Erfolgsmeldung „LASAR ÜÄ ze" und das Integer-BASIC-Prompt-Zeichen. Das Betriebssystem wird nicht selbstständig geladen, sondern muß durch „PR#6" gestartet werden (wohl dem, der über eine System-Master-Diskette mit Applesoft verfügt).

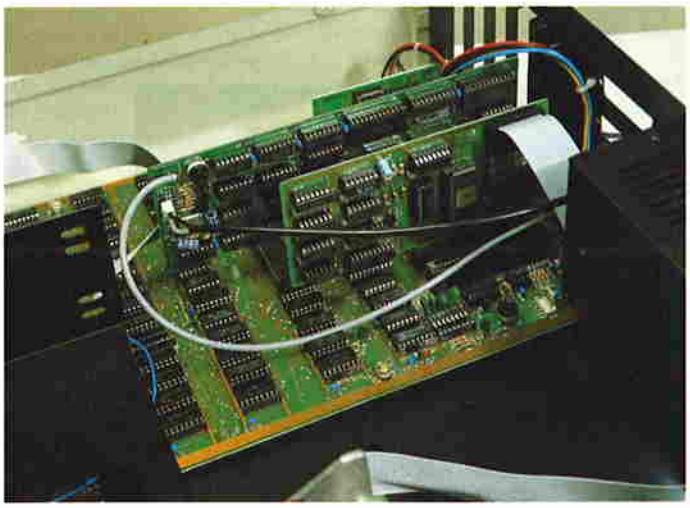
Der BASIC-Interpreter befindet sich auf einer ROM-Karte, die in Slot 0 steckt; eine Applesoft-Karte kann nicht (!) bezogen werden. Dies bedeutet, daß ProDOS nicht gestartet werden kann, solange keine reine Applesoft-Firmware vorhanden ist, da ProDOS hier erfreulicherweise nicht durch den Controller zum erliegen kommt, sondern den Interpreter moniert.

Die Tastatur

Die deutsche Tastatur von Multitech wurde in erster Linie für die Programmierung unter Applesoft und DOS 3.3 entwickelt. So sind 12 Funktionstasten mit Befehlen dieser Systeme vordefiniert. Dabei führt z.B. die Taste F12 den HOME-Befehl sofort aus (d.h. RETURN wird übertragen), während der RUN-Befehl ohne RETURN übertragen wird. Dies erscheint wohl einsichtig, bedarf jedoch einiger Gewöhnung. Darüber hinaus erlaubt eine Alternate-Taste den Zugriff auf weitere 52 BASIC-Schlüsselwörter mit einem Tastendruck. Diese Option kann nur nach langem Umgang mit dieser Tastatur gewinnbringend genutzt werden. Als sehr praktisch erweist sich die Möglichkeit, 10 Tasten mit eigenen Kommandos zu belegen, solange der Computer nicht abgeschaltet wird.
Außer einem separaten Ziffernblock sind noch einige Tasten enthalten, die spezielle ESC-Sequenzen oder Ctrl-Funktionen generieren oder automatische Zeilennumerierung ermöglichen. Trotz der nicht gerade ergonomisch angelegten RETURN-Taste weiß jeder diese Tastatur zu schätzen, der schon einmal versucht hat, dem alten Europlus-Keyboard ein „ö" zu entlocken.



Die Groß-/Kleinschreibung wird von der Hard- und Firmware des Lasar unterstützt, unter Applesoft können Kleinbuchstaben jedoch nur mit Hilfe der 80-Zeichen-Karte eingegeben werden.



Controller und 80-Zeichenkarte

Der Controller beweist seine Kompatibilität, wie schon oben erwähnt, beim Booten des ProDOS-Systems und arbeitet auch, wenn bereits Applesoft in die 16K-Karte geladen wurde. Damit ist eine ausreichende Kompatibilität gewährleistet.
Die 80-Zeichenkarte verfügt über einen Softswitch und ist im großen ganzen videx-kompatibel. So läuft der entsprechend konfigurierte Merlin-Assembler wie auch die Betriebssysteme UCSD und CP/M ohne Schwierigkeiten. Zum Test lag eine Karte mit amerikanischem Zeichensatz vor. Laut Angabe der Firma soll jedoch auch eine deutsche oder wahlweise umschaltbare Version lieferbar sein.

Kompatibilität

Einleitend kann gesagt werden, daß keine Kompatibilitätsprobleme auftraten. Nach Laden des Applesoft-Interpreters liefen alle BASIC-Programme anstandslos.
Die Betriebssysteme UCSD und CP/M konnten ohne Schwierigkeiten gestartet werden. Das Arbeiten mit dem Big-Mac- bzw. Merlin-Assembler und der Applewriter-Version mit Videx-Preboot-Diskette gestaltet sich auf Grund der Groß-Kleinschreibtastatur bedeutend komfortabler.
Die Probleme mit ProDOS können wohl nur bei illegalen Nachbauten umgangen werden; bei Kompatiblen bleibt zu guter Letzt der legale Weg, die benötigte Firmware bei einem „durch persönliches Band verbundenen Bekannten" mit Original-Apple nachbrennen zu lassen. Unter diesen Umständen dürfte der Lasar II ze so kompatibel sein wie der Apple II Europlus zu seinem amerikanischen Bruder.

Fazit
Wenn man von der recht bescheidenen „Paperware" absieht, stellt der Lasar II ze eine echte Alternative zu den bedeutend teureren Apples dar. Wer ohne Betreuung der Firma Apple leben kann und in Kauf nimmt, etwa zukünftige Betriebssysteme nicht benutzen zu können, weil zunächst eine Prüfsumme der kompletten Firmware gebildet wird, ist mit dem Lasar sicher nicht schlecht beraten.
Das beschriebene Gerät wurde ca. drei Wochen lang zu Testzwecken eingesetzt. In diesem Zeitraum konnten keine technischen Mängel festgestellt werden.

Ausgabe und **Eingabe** mit
TYPETERM®
im Slot Ihres
APPLE II/IIe
Das bedeutet: Computertextverarbeitung von der Schreibmaschinentastatur! Steckerfertig ohne Umbau.
TYPETERM-Interface **DM 479,–** incl. MWSt.
für alle BROTHER-Typenradschreibmaschinen
Paketpreis: **DM 1348,–**
Schreibmaschine CE-51 mit TYPETERM

brother
QUALITÄT AUS ERSTER HAND.
Alle Maschinen günstig, z.B.:
CE-61 ..... DM 1308,–
EM-80 ..... DM 1608,–
EM-100 ..... DM 2643,–
TYPETERM-Kit für CE-50 ..... DM 468,–
CE-25 mit TYPETERM ..... anfragen
TYPETERM – ein starkes Interface für starke Maschinen! Alle Cursor- und Ctl-Befehle. 2k ROM auf der Karte f. DOS, PRODOS, CP/M, PASCAL. Alle Features: Hoch-/Tiefstellen, autom. Unterstreichen, var. Zeichen- u. Zeilenabst., autom. Papierzuführung usw. Ausführl. Handbuch vorab: 10,– DM auf Konto 4770-306 PGiroA Han (Anrechnung).
TYPETERM ein Produkt von
interkom electronic
Kock & Mreches GmbH
Postf., 3004 Isernhagen 4
Telefon 05139-87393

Typenradschreibmaschine Brother CE 50

getestet von Rainer Gerdes

Allgemeines: Im November 1984 kaufte ich mir bei der Firma Intercom GmbH in Isernhagen eine elektronische Schreibmaschine vom Typ Brother CE 50 mit entsprechendem Interface für meinen Apple II Plus. Der Preis lag zu dieser Zeit bei DM 1298,- und erschien mir bei Preisvergleichen mit anderen Anbietern als günstig. Die räumliche Nähe des Verkäufers gab wegen vieler schlechter Erfahrungen mit anderen Computeranbietern den Ausschlag. Bei der CE 50 handelt es sich um eine portable elektronische Schreibmaschine mit folgenden Zusatzausstattungen:
– automatischer Papiereinzug,
– Halbschritt vor- und rückwärts,
– Indizierung und Exponentiation.
Mit der Maschine und dem Interface wird noch eine 50seitige (!) Beschreibung mitgeliefert. Darin sind die Möglichkeiten der Maschine beschrieben: die Installierung und der erste Test, die verschiedenen Betriebsarten sowie das Verhalten bei den verschiedenen Apple-Betriebssystemen DOS 3.3, CP/M und UCSD. Am Ende sind Subroutinen aufgelistet und dokumentiert, um die Maschine auch unter CP/M und UCSD als Eingabeeinheit verwenden zu können. Diese Subroutinen können auch vom Anbieter gegen eine Gebühr von DM 25,- pro Betriebssystem auf Diskette bezogen werden. Das Interface hat keine Standardschnittstelle, sondern nutzt den firmenspezifischen Eingang der Schreibmaschine. Das sollte in Betracht gezogen werden, wenn man in absehbarer Zeit einen Systemwechsel vornehmen will.

Betriebserfahrungen: Die Maschine wurde bis jetzt mit den beiden Textverarbeitungssystemen Applewriter II und Wordstar getestet. Dabei tauchten keine Schwierigkeiten auf. Die Sonderfunktionen sind über Ctrl-Zeichen zu erreichen und werden von der Maschine anstandslos ausgeführt. Zu den Sonderfunktionen gehören z.B. Unterstreichung, automatischer Papiereinzug, Wahl des Zeichenabstandes, Wahl des Zeilenabstandes, Halbschritt usw.
Die Besonderheit des Interfaces kommt aber erst bei dem Textverarbeitungssystem Wordstar zum Tragen, nämlich die Benutzung der Schreibmaschine als Eingabetastatur. Nach dem Patchen des CP/M-Betriebssystems mit der im Handbuch beschriebenen Routine kann der Wordstar normal mit „Install" angepaßt werden. Es ist nur darauf zu achten, daß beim Driver-Menu die Möglichkeit CP/M "List" Device gewählt wird und daß sich das Programm STAT auf der Diskette befindet. Mit zwei Befehlen ist dann vor dem Starten des Wordstar die Schreibmaschine als Ein- und Ausgabeeinheit aktiviert. Dabei sind Druckwerk und Tastatur getrennt. Man kann also Texte, ohne einen Ausdruck zu bekommen, direkt in den Rechner schreiben. Durch die vorhandenen Zusatztasten sind auch die ESC- und Ctrl-Funktionen erreichbar. Die Schreibmaschine macht also eine Zusatztastatur, auf die man beim Apple II+ wohl nicht verzichten kann, überflüssig, wodurch gut DM 500,- gespart werden können. Leider ist diese Eingabemöglichkeit auf den Wordstar beschränkt. Der Applewriter ist so angelegt, daß nur von der Apple-Tastatur aus zugegriffen werden kann.

Erfahrungen mit dem Anbieter: Die anfängliche Skepsis dem Anbieter gegenüber rührt aus einem reichen Erfahrungsschatz mit anderen Computerverkäufern her und war auch ein Punkt meiner Kaufentscheidung (s.o.). Diese Skepsis wich aber bald und machte einem positiven Urteil Platz. Niemand ist fehlerlos und so wurde auch meine Schreibmaschine von einer Kinderkrankheit befallen. Was in Basic einwandfrei funktionierte, nämlich Ein- und Ausgabe von der Schreibmaschine, wollte nach dem Patchen des CP/M-Systems nicht funktionieren. Nach mehreren Versuchen schickte ich erst einmal die Diskette zurück (ich hatte mich entschlossen, die 25.- DM zu investieren) und bekam nach 24 Stunden Ersatz. Als es immer noch nicht funktionieren wollte, wurde das EPROM auf der Platine ausgetauscht. Das ging alles sehr schnell und reibungslos.

Nachteile der Maschine: Kein Licht ohne Schatten. Der erste Kritikpunkt ist die Geschwindigkeit der Maschine. 13 Zeichen/Sekunde ist natürlich sehr langsam, und man braucht etwas Geduld. Die Schreibmaschine wird von CP/M nur in Slot 2 entdeckt. Ist die Schreibmaschine in Slot 2 eingesteckt, so meldet sich CP/M nur, wenn die Maschine in Betrieb ist. Wurde der Rechner ausgeschaltet, so muß auch die Schreibmaschine ausgeschaltet werden. Andernfalls bleibt das Schreibwerk in der Mitte der Maschine stehen und CP/M bootet nicht mehr. Es müssen also immer beide Maschinen gleichzeitig gestartet werden.

Fazit: Eine schöne Maschine mit kleinen Fehlern, die aber trotzdem sehr empfehlenswert ist. Die gleiche Empfehlung gilt für den Anbieter.

Brother CE 50

Editierhilfen für den Apple

getestet von Franz-Josef Hüskens

Jeder, der mit dem Apple ernsthaft programmieren will, hat mit Sicherheit (und Bedauern) bereits festgestellt, daß dieser Computer von Hause aus nur mit sehr mageren Editierfähigkeiten ausgestattet ist. Das Programmieren in BASIC wird fast zur Tortur, da bei der Korrektur einer Zeile mit „wilden" Tastensequenzen gearbeitet werden muß und dabei die Fehleranfälligkeit reichlich hoch ist.
Dies macht es dann früher oder später erforderlich, ein Programm zu erstehen, das bei der Berichtigung und Veränderung von Programmzeilen hilft und vielleicht auch noch sonstige Programmierunterstützung bietet. Hier sollen nun zwei Programme vorgestellt werden, von denen jedes für sich eine große Hilfe darstellt, die jedoch zur Freude des Programmierers auch gemeinsam ihre Arbeit verrichten können.

Global Programm Line Editor

Die Utility GPLE ist ein allgemeines Programmzeilen-Bearbeitungsprogramm. Es ist sowohl unter DOS 3.3 als auch unter ProDOS einsetzbar und wird auf einer Diskette geliefert, auf der sich beide Programmversionen befinden. Wird diese Diskette gebootet, wird automatisch ProDOS geladen und GPLE installiert. Will man unter DOS 3.3 mit GPLE arbeiten, so muß man mit einer „normalen" Diskette booten und dann GPLE laden.
Unter DOS 3.3 gibt es dafür drei Möglichkeiten:
1. zwischen die DOS-Puffer,
2. in die 16K-Karte (mit einer zweiten Sprache),
3. mit DOS in die 16K-Karte.
Für diese drei Speicherbereiche stehen auf der Diskette drei Programmversionen zur Verfügung. Als weiteres Programm befindet sich auf der Diskette ein DOS-Mover; dieser wird bei der dritten Möglichkeit benötigt.
Bevor man GPLE lädt, kann man es für den eigenen Gebrauch „einstellen" (konfigurieren). Dabei können folgende Möglichkeiten gewählt werden:
1. eine von sechs direkt unterstützten 80-Zeichen-Karten (für weitere Karten sind Treiber-Programme auf der Diskette vorhanden),
2. ob die 80-Zeichen-Karte direkt nach dem Laden von GPLE eingeschaltet werden soll,
3. Simulierung eines Tastaturpuffers,
4. Groß- und Kleinschreibung.
Auf Schwierigkeiten, die die Benutzung des Tastaturpuffers hervorrufen kann, wird im Handbuch ausdrücklich hingewiesen.

Die Benutzung von GPLE

Das wohl wichtigste Kommando ist Ctrl-E. Danach erhält man die Möglichkeit, eine oder mehrere Programmzeilen zu editieren oder in einem Zeilenbereich bzw. im gesamten Programm Wörter zu suchen und/oder zu ersetzen.
Es wird die gewünschte Zeile gelistet und zur Bearbeitung zur Verfügung gestellt. Beim Editieren eines Zeilenbereichs wird die jeweils zur Bearbeitung anstehende Zeile angezeigt.
GPLE bietet folgende Editierhilfen:
– Positionierung des Cursor (zum Anfang oder zum Ende der Zeile oder auf ein bestimmtes Zeichen);
– Einfügen und Löschen beliebiger Zeichen (auch Ctrl-Zeichen);
– Löschen von ganzen Zeichenketten;
– „Packen" einer Zeile, d.h. alle Leerzeichen (Blanks), die nicht in Anführungszeichen stehen, werden entfernt;
– Umwandlung des Zeichens unter dem Cursor in einen Klein- bzw. Großbuchstaben;
– man kann aber auch alle Änderungen wieder rückgängig machen.
Neben dem Editieren von Programmzeilen ist es auch möglich, mit Ctrl-W Befehle, die im direkten Modus eingegeben und noch nicht mit RETURN bestätigt wurden, zu verbessern.
Der große Vorteil, den GPLE dabei bietet, ist die Akzeptierung einer Zeile unabhängig von der momentanen Cursorposition, d.h. bei einer Änderung oder Verbesserung, die am Anfang einer Zeile nötig war, muß der Zeilenrest nicht mit dem Cursor „überfahren" werden, um ihn zu übernehmen. An jeder beliebigen Stelle der Zeile kann RETURN gedrückt werden, und die Zeile wird so übernommen, wie sie auf dem Bildschirm zu sehen ist.

Dies ist aber noch nicht alles, was GPLE bietet!

Terminals an größeren Computern haben meist neben der Tastatur noch Funktionstasten, die mit häufig benötigten Befehlen oder Befehlssequenzen programmiert werden können.

Durch GPLE wird der Apple auch mit solchen „programmierbaren" Tasten ausgerüstet. Zur Befehlsausführung müssen dann zwei Tasten, nämlich die ESC-Taste und die „programmierte" Taste betätigt werden.

Um eine Taste zu programmieren, wird mit Ctrl-E zuerst in den Editiermodus umgeschaltet, dann ESC und die zu definierende Taste eingegeben und zum Schluß der oder die gewünschten Befehle programmiert. Dabei stehen insgesamt 1152 Bytes zur Verfügung.

Eine so definierte ESC-Tabelle kann mit dem entsprechenden Hilfsprogramm auf Diskette gespeichert bzw. auf dem Drucker ausgegeben werden. Die zwei Hilfsprogramme werden mitgeliefert. Natürlich ist es auch möglich, die Tabelle seitenweise auf dem Bildschirm anzuzeigen.

Außerdem befinden sich auf der Diskette noch drei Programme, mit denen FID, MUFFIN und RENUMBER so geändert werden, daß sie mit dem verschobenen DOS arbeiten können.

Zusammen mit der Diskette erhält man ein 38 Seiten starkes Handbuch, das nicht nur die Befehle detailliert und anhand von Beispielen beschreibt, sondern auch die Benutzung aller auf der Diskette mitgelieferten Programme erklärt. Zusätzlich enthält das Handbuch noch einen Teil mit verschiedenen allgemeinen Tips zur besseren und übersichtlicheren Programmierung und Tips zur Anwendung von GPLE.

DOUBLE TAKE

DOUBLE TAKE kann allein arbeiten; es kann jedoch auch zusammen mit GPLE geladen werden, und zwar an die gleichen Stellen (s.o). Man kann bis zu vier Module laden (die Module werden vom Hersteller mit Farben bezeichnet), wobei das fünfte automatisch eingelesen wird.

Die Module enthalten folgende Funktionen:
– Das rote Modul enthält eine neue LIST- und CATALOG-Routine. Beim Listen eines Programms gibt es nun zwei Möglichkeiten: das altbekannte Applesoft-Listing und ein neues Listing, bei dem alle Befehle, die in einer Programmzeile stehen können, in eine neue Bildschirmzeile geschrieben werden. Sowohl beim CATALOG als auch beim Listen ist es möglich, mit den Pfeiltasten die Ausgabe vorwärts und rückwärts zu steuern.
– Das orange Modul enhält diverse Monitor-Hilfen. Neben einem Hex/ASCII-Dump, einer Disassemblierung und einer Hexadezimal/Dezimal-Umrechnung ist es auch möglich, Monitor-Kommandos aus dem Applesoft-Direktmodus heraus einzugeben.
– Das blaue Modul gestattet es, zwei Applesoft-Programme zu mergen (mischen), Zeilen umzunumerieren (beides entspricht dem Applsoft-Programm RENUMBER) und Zeilen automatisch zu numerieren (entspricht dem AUTO-Befehl beim Integer-BASIC).

– Das gelbe Modul bietet:

1. Programmstatistik (Zeiger und Speicherbereiche des BASIC-Programms);
2. 2-Byte-PEEK (entspricht PEEK (adr) + PEEK (adr + 1) * 256);
3. Liste der Variablen eines Programms, alphabetisch sortiert mit den Nummern der Zeilen, in denen sie auftreten;
4. Anzeige der einfachen Variablenwerte eines Programms (keine Felder);
5. „Cursor-Ersatz" (der Cursor kann neu definiert werden);
6. Sichtbarmachen von Ctrl-Zeichen.

Zusätzlich gibt es ein Modul mit „Bildschirmschaltern" zum einfachen Umschalten zwischen Text und Grafik (Lores, Hires)

Alle Komandos belegen die oberste Reihe der Tastatur und werden mit Ctrl-F aufgerufen. In der Anzeige erscheint dann „FUNCTION?". Nun muß die mit der gewünschten Funktion belegte Taste gedrückt werden, worauf gegebenenfalls weitere „Hilfs"-Fragen erscheinen. Als Gedächtnisstütze wird eine Kommando-Tafel mitgeliefert, die man hinter die oberste Tastenreihe stecken kann.

Das Programmsystem kann wegen der modulweisen Lademöglichkeit

den eigenen Wünschen und Bedürfnissen entsprechend zusammengestellt werden. Das dazu benötigte Programm erlaubt nicht nur die Auswahl der Module in der normalen Länge, sondern bietet die Teilprogramme in einer verkürzten Version an (bis auf das gelbe Modul) und zeigt, woraus die verkürzten Module bestehen.

Alle fünf Module zusammen benötigen in der normalen Form 8K Speicherplatz. Bei entsprechender Speicherung (DOS und GPLE in der 16K-Karte, DOUBLE TAKE darunter) verbleiben ungefähr 38K.

Die Benutzung von DOUBLE TAKE wird in einem 20seitigen Handbuch gut beschrieben. Dabei wird auch auf die Unterschiede im Aufbau der Tastatur des Apple II Plus und des Apple IIe eingegangen. Zusätzlich erhält man die bereits oben erwähnte Kommando-Tabelle.

Zur Zeit gibt es in Deutschland noch die Version 1.2 von DOUBLE TAKE. Diese Version arbeitet nicht mit GPLE zusammen! Wollen Sie beide Programme gemeinsam benutzen, so müssen Sie beim Kauf unbedingt die DOUBLE-TAKE-Version 2.0 verlangen. Diese Version unterstützt auch die 80-Zeichen-Karte, wobei 11 verschiedene Kartentreiber zur Verfügung stehen. Beide Programme werden von der Beagle Bros Inc. verlegt und sind in Deutschland bei verschiedenen Importeuren erhältlich. GPLE kostet ca. DM 180,- (und ist seinen Preis wert!), DOUBLE TAKE ca. DM 140,-.

Lernprogramme von INTUS

getestet von
Dr. Jürgen B. Kehrel

Die Firma INTUS Lern-Systeme AG bietet eine große Zahl von Programmen an, die Ihren Apple zum geduldigen und anspruchsvollen Lehrer werden lassen. Alle Programme sind in deutscher Sprache und auf einem Apple IIe oder IIc lauffähig. Der ältere Apple II Plus ist wegen fehlender Kleinschreibung und der eingeschränkten Tastatur (Umlaute) nur bedingt einsatzfähig, z.T. gar nicht. Die Disketten sind kopiergeschützt. Zu Testzwecken wurde dem Peeker eine Auswahl der Programme zur Verfügung gestellt, die nachfolgend charakterisiert werden.

Maschineschreiben wie der Blitz
In 20 Lektionen für Anfänger und Fortgeschrittene wird ein kompletter Lehrgang ohne zusätzliches Lehrbuch geboten. Die einzelnen Tasten (inkl. Groß- und Kleinschreibung), Wörter und Sätze werden geübt. Ihre Genauigkeit und Schnelligkeit wird getestet, so daß Sie gezielt üben können, um Ihre Fähigkeiten zu vervollkommnen. 1 Applesoft-Diskette, DM 188,–. Für Apple II Plus ungeeignet.

Computer-Simulator
Dieses Programm hilft Ihnen, die Arbeitsweise eines echten Computers zu verstehen. Ein vereinfachter Rechner mit nur 4-Bit-Prozessor, 20 Speicherplätzen und 10 Befehlen läßt Sie in sein Innenleben (Ein- und Ausgabe, Zentraleinheit mit Akkumulator, Programmzähler und Instruktionsregister) schauen. Für Schüler ab etwa 6. Schuljahr. 1 Applesoft-Diskette plus Manual, DM 146,-.

Applesoft-Basic-Tutorial
Mit diesem Tutorial und 9 Seiten Text werden Sie lernen, in Applesoft zu programmieren. In 12 Lektionen wird der ganze Befehlssatz des Apple (außer der hochauflösenden Grafik) mit Ihnen durchgearbeitet. Zwei Anwendungstutorials zeigen Ihnen kommerzielle, medizinische, statistische und schulische Beispiele sowie Spiele. 2 Applesoft-Disketten, DM 295,-.

Deutsche Grammatik mit Spaß
Sie besteht aus einer umfangreichen Serie mit zur Zeit 15 Disketten zu Rechtschreibung, Zeichensetzung, Wortarten, Verben, Zeitenfolgen, Adjektiven und Adverbien, Satzgliedern, Haupt- und Nebensätzen, Transformationen, Wortbedeutungen und stilistischen Problemen. Neben Regeln und Erklärungen stehen zahlreiche Übungen, die Sie zwischen 2 und 5 Stunden pro Diskette beschäftigen. Arbeitstempo und Schwierigkeitsgrad sind wählbar. 15 Pascal-Disketten, je DM 165,-. (Apple-Pascal nicht erforderlich.)

Algebra, leicht verständlich
Die Übungsprogramme dienen zur Vertiefung des Elementar-Algebra-Stoffes. Interaktiv werden Sie in mehreren Schwierigkeitsebenen zur selbständigen Lösung geleitet. Bei falschen Lösungen werden Beispiele aufgezeigt und die Lösungsschritte nachvollzogen. Die Programme sind für die Sekundarstufe 1 und 2 bestimmt. 4 Applesoft-Disketten (Preis?).

Lesen wie der Blitz
Das Programm dient dazu, Ihre Lesegewohnheiten zu verbessern und gleichzeitig Ihr Lesetempo zu steigern. Ihre Konzentrationsfähigkeit wird gefördert, indem Ihnen Zahlen, Wörter oder Sätze für kurze Zeit gezeigt werden, die Sie niederschreiben sollen. Sie erhalten dann die Antwort, ob es richtig war. Durch regelmäßige Übung (täglich 30-60 Minuten) steigern Sie Ihre Geschwindigkeit. 1 Applesoft-Diskette, DM 115,–.

Erscheinungs- und Anzeigenschlußtermine für **peeker**

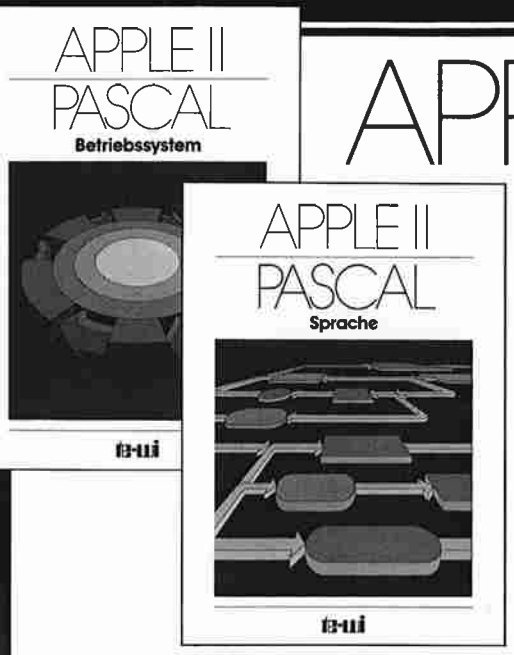
Ausgabe	Erscheinungstermin	Anzeigenschluß
7	24. 6. 85	24. 5. 85
8	22. 7. 85	21. 6. 85
9	26. 8. 85	26. 7. 85
10	23. 9. 85	23. 8. 85
11	21. 10. 85	20. 9. 85
12	25. 11. 85	25. 10. 85

. . . in eigener Sache:

die Nachfrage nach dem vergriffenen Heft 1/84 ist groß. Ab sofort können Sie eine Heftkopie direkt beim Verlag bestellen.

Preis für das Inland:
DM 10,– inkl. Versandkosten

für das Ausland:
DM 12,– inkl. Versandkosten
eventuelle Luftpostzuschläge zzgl.



APPLE II PASCAL

Betriebssystem und Sprache

Erstes deutsches Referenzwerk sämtlicher Befehle und Systemroutinen von Apple II Pascal – mit Addendum einschließlich Version Pascal 1.2!

Gültig für Apple II, II Plus, IIe einschließlich der 128K/80 Zeichen-Konfiguration.

Betriebssystem kommentiert ausführlich und in Deutsch Funktion und Benutzung der fast 60 Systemroutinen des Apple II Pascal Betriebssystems.

Sprache ist das vollständige, deutsche Referenzwerk der „Apple Pascal"-Programmiersprache mit u. a. Informationen über professionelle Pascal-Programmierung, Turtlegraphics, Programmbibliothek etc.

In Vorbereitung: Addendum Pascal 1.2, ein Zusatz zum Buch „Betriebssystem" für 1.2-Benutzer in Deutsch.

„Nach Unterlagen von Apple Deutschland hergestellt"

Apple II Betriebssystem,
272 Seiten, DM 49,–

Apple II Sprache,
216 Seiten, DM 39,–

Pascal 1.2 Addendum, etwa 100 Seiten,
DM 36,–

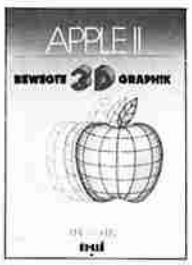
te-wi Verlag GmbH
Theo-Prosel-Weg 1
8000 München 40

Weiterführende Literatur...

APPLE II – Anwenderhandbuch
(L. Poole)
Erst mit Hilfe dieses Leitfadens werden Sie Ihren Apple II erfolgreich einsetzen, denn Text und Bildmaterial gehen weit über das hinaus, was herstellerseitig an Literatur angeboten wird.
416 Seiten, Softcover, DM 56,–

LOGO – Jeder kann programmieren
(Daniel Watt)
Buch des Jahres in den USA. Für die Computer C64, ATARI, APPLE II, IBM-PC und TI-99.
Hochwertiges Textbuch für Logo-Kurse für zu Hause und im Lehrbereich.
A4, DM 59,–

APPLE II PASCAL – Eine praktische Anleitung
(A. Luehrmann, H. Peckham)
Unentbehrlich für alle, die die Programmiersprache PASCAL lernen wollen und Zugang zu einem Apple-Computer haben.
544 Seiten, Softcover, DM 59,–



APPLE II – Bewegte 3D-Graphik
(Phil Cohen)
Selbstentworfene Graphiken und Diagramme – animiert oder als Standbilder – eben oder räumlich: alle erforderlichen BASIC-Programme mit Erklärung finden Sie in diesem Buch.
ca. 190 Seiten, Softcover, DM 49,–

Computer für Kinder
(Sally Greenwood Larson)
Ein Buch für Kinder, ihre Lehrer und Eltern.
„Computer für Kinder" richtet sich an Kinder im Alter von 8 bis 13 Jahren, für deren Interesse an Computern dieses Buch bewußt geschrieben wurde.
Unterhaltsam und leicht verständlich,
A4 quer, Fadenheftung, DM 29,80

Apple Maschinensprache
Für BASIC-Programmierer der einfachste Zugang zur Muttersprache des Apple. Wesentlich schnellere Maschinenprogramme, direkte Manipulation des Mikroprozessors 6502 im Apple – als Brücke dorthin benötigt dieses Buch nur die drei BASIC-Befehle, POKE, CALL, PEEK. D. Inman/K. Inman. DM 49,–

tm 4153

Noch im Programm:
6502 – Programmieren in Assembler DM 59,–
VisiCalc, 50 Programme auf Diskette, DM 79,–

In Vorbereitung:
Macintosh Programmier-Handbuch DM 59,–

intelligente Peripherie für Ihren Apple®-Rechner in Profi-Qualität, made in Germany

- Laufwerke
- RAM-Karten
- ★ Controller
- komplette Sub-Systeme

Information und Verkauf über den Fachhandel

erphi electronic GmbH, Dammweg 3, 8011 Großhelfendorf, tx 528 021 erphi d